

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ГАБРОВО
Факултет „Електротехника и електроника”

**Разработване и изследване на хардуерно
базирани изкуствени невронни мрежи от тип DNN
за разпознаване и класификация**

маг. инж. Станко Петров Станков

А В Т О Р Е Ф Е Р А Т

на
ДИСЕРТАЦИЯ

за придобиване на образователна и научна степен „доктор”

Област на висше образование: **5. Технически науки**
Професионално направление: **5.2. Електротехника,
електроника и автоматика**
Специалност: **Елементи и устройства на автоматиката и
изчислителната техника**

Научен ръководител: доц. д-р инж. Стефан Иванов Иванов

Габрово
2022

**Разработване и изследване на хардуерно базирани
изкуствени невронни мрежи от тип DNN за
разпознаване и класификация**

маг. инж. Станко Петров Станков

А В Т О Р Е Ф Е Р А Т

на
ДИСЕРТАЦИЯ

за присъждане на образователна и научна степен „доктор”

Научен ръководител: доц. д-р инж. Стефан Иванов Иванов

Рецензенти:

проф. дтн инж. Чавдар Иванов Дамянов

доц. д-р инж. Станимир Йорданов Йорданов

Габрово
2022

Дисертационният труд съдържа 181 страници. Структуриран е в увод, пет глави, заключение, приноси в дисертационния труд и библиография. Включва 153 фигури и 23 таблици. Библиографията съдържа 155 източника, от които 2 на кирилица и 154 на латиница.

Дисертационният труд е обсъден и насочен за защита на заседание на разширен катедрен съвет на катедра „Автоматика, информационна и управляваща техника”, факултет ”Електротехника и електроника” на Технически университет – Габрово,

Дисертационният труд е разработен по време на редовна докторантура в катедра „Автоматика, информационна и управляваща техника” на Технически университет – Габрово.

Номерацията на формулите, фигурите и цитираната литература съответства на тази в дисертационния труд.

Защитата на дисертационната работа ще се състои на от ч. в зала, корпус..... на Технически университет – Габрово.

Материалите по защитата са на разположение за интересуващите се в кабинет 3209, корпус 3 на Технически университет – Габрово.

Автор: маг.инж. Станко Петров Станков

Заглавие : Разработване и изследване на хардуерно базирани изкуствени невронни мрежи от тип DNN за разпознаване и класификация

Тираж: 30бр.

Печатна база при ТУ-Габрово

ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД

АКТУАЛНОСТ НА ПРОБЛЕМА

Изкуственият интелект в съвременното общество не само може да подобни човешкия живот в някои случаи дори може да има животоспасяваща функция. Има много технологични компании, които инвестират и изследват начини, по които изкуственият интелект може да помогне за подобряване на начин на живот.

Машинното обучение е под-област на изкуствения интелект, при което терминът се отнася до способността на системите да намират самостоятелно решения на проблеми чрез разпознаване на модели в бази данни.

Изкуствени невронни мрежи са изчислителни модели които имат уникалната способност да извличат информация от неточни или сложни данни, за да намират модели на поведение и да откриват тенденции, които са твърде неясни за човешкия мозък или за други компютърни техники.

Те имитират процесите на човешкия мозък, които са изградени от неврони - основният градивен елемент на предаването на информация, както на човека, така и на невронната мрежа.

Дълбоките невронни мрежи (DNN- Deep Neural Network), от своя страна, осигуряват несравнима точност и производителност във все по-широк спектър от индустриални приложения като разпознаване на изображения, обработка на естествен език и други сложни проблеми , сред които управление на автономни превозни средства. Въпреки огромните подобрения на резултатите в сравнение с по-старите алгоритми за машинно обучение, DNN са много взискателни по отношение на изчисленията и изискват обучение с масивни набори от данни, отнемащи големи количества време.

Конволюционните невронни мрежи (CNN) представляват алгоритми за дълбоко обучение, които могат въз основа на входни изображения да откриват различни обекти в тях.

ЦЕЛ И ЗАДАЧИ НА ДИСЕРТАЦИЯТА

Целта на дисертационния труд е, да се изследват и разработят подходи и алгоритми на базата на изкуствени невронни мрежи за осигуряване сигурността на хората в различните области от ежедневието им живот.

При разработването на дисертационния труд се дефинират следните основни задачи:

След като се разгледат съществуващи модели на невронни мрежи и алгоритми за трениране, като задачи да се предложат модификации за алгоритмите и архитектурите на мрежите, които да бъдат използвани при работа с хардуерно базирани модули.

Да се представи решаването на проблеми свързани със сигурността, с помощта на DNN мрежи - откриване на различни видео опасности като насилие, оръжие и пожар, както и разпознаване на различни жестове с ръка.

Да се направи сравнителен анализ на работата на различни типове невронни мрежи за решаване на поставените задачи.

Обучените невронни мрежи да се имплементират в съответните хардуерни модули с цел тяхното практическо приложение.

ПРАКТИЧЕСКО ПРИЛОЖЕНИЕ

Част от резултатите от дисертационния труд ще се използват и при обучение на студенти по дисциплината „Интелигентни сензори и системи” в образователно-квалификационна степен „бакалавър”, специалност „Автоматика, информационна и управляваща техника” в Технически университет - Габрово.

Резултатите от дисертационния труд могат да нямерят приложение в решаването на различни проблеми свързани със системите за сигурност като откриване на оръжие, разпознаване на жестове с ръка и детекция на пожари.

АПРОБАЦИЯ

Вследствие на извършената научно-изследователска работа по дисертационния труд са

изнесени 10 доклада в научни конференции с международно участие и са публикувани 2 статии в списания. Четири от научните доклади са изнесени на ежегодното провеждане на международни научни конференции Унитех в Технически университет – Габрово. Един доклад е изнесен на международната научна конференция 20th International Symposium on Electrical Apparatus and Technologies (SIELA), един доклад на конференцията - VII international scientific conference "Engineering. Technologies. Education. Safety", един доклад на международната конференция „Изкуствен интелект и Е-лидерство”.

Една от публикациите е самостоятелна, а останалите са в съавторство с научния ръководител. Една от публикациите е в съавторство с учени от Полша.

Три от научните доклади са реферирани в базата данни на Scopus.

Направените по дисертацията доклади са цитирани в пет научни публикации от други автори, реферирани в базата данни на Google Scholar.

КРАТКО СЪДЪРЖАНИЕ НА ДИСЕРТАЦИЯТА

ГЛАВА I. Изкуствени невронни мрежи от тип DNN. Основни характеристики и приложение

Машинно обучение

Машинното обучение е приложение на изкуствен интелект, което предоставя на системите възможност за автоматично учене и усъвършенстване от опит, без да бъде изрично програмирано. Машинното обучение се фокусира върху разработването на компютърни програми, които могат да имат достъп до база данни и да ги използват за самообучение [151]. Процесът на обучение започва с наблюдения или данни, като примери, директен опит или инструкции, за търсене на по-добри решения в бъдеще въз основа на предаваната информация. Основната цел е да се позволи на компютрите автоматично обучение без човешка намеса.

Изкуствени невронни мрежи

Изкуственият интелект в съвременното общество не само може да подобри човешкия живот в някои случаи дори спасява животи. Има много технологични компании, които инвестират и изследват начини, по които изкуственият интелект може да помогне за подобряване на човешкия живот.

Машинното обучение е под-област на изкуствения интелект, при което терминът се отнася до способността на системите да намират самостоятелно решения на проблеми чрез разпознаване на модели в бази данни.

Изкуствени невронни мрежи са изчислителни модели, които имат уникалната способност да извличат информация от неточни или сложни данни, за да намират модели на поведение и да откриват тенденции, които са твърде неясни за човешкия мозък или за други компютърни техники.

Те имитират процесите на човешкия мозък, които са изградени от неврони - основният градивен елемент на предаването на информация, както на човека, така и на невронната мрежа.

Дълбоките невронни мрежи (DNN- Deep Neural Network), от своя страна, осигуряват несравнима точност и производителност във все по-широк спектър от индустриални приложения като разпознаване на изображения, обработка на естествен език и други сложни проблеми, сред които управление на автономни превозни средства. Въпреки огромните подобрения на резултатите в сравнение с по-старите алгоритми за машинно обучение, DNN са много взискателни по отношение на изчисленията и изискват обучение с масивни набори от данни, отнемат много време.

Конволюционната невронна мрежа (CNN) е алгоритъм за дълбоко обучение, който може да взема входни изображения и да открива различни обекти в тях.

Напредъкът в изчислителната техника, наличието на огромни количества данни и нови алгоритми доведоха до големи пробиви в изкуственият интелект през последните години. Изкуственият интелект има различни приложения в днешното общество. Той става от съществено значение за днешното време поради неговата възможност да разрешава сложни проблеми по ефективен начин в множество индустрии, като здравеопазване, развлечения, финанси, образование и други. Приложения, които използват изкуствен интелект, са следните: автономни автомобили, интелигентни къщи, откриване на аномалии, роботика, автоматизация, качествен контрол и др.

Хардуерни ускорители за работа с изкуствени невронни мрежи използвани в дисертационния труд

За работа с изкуствени невронни мрежи в дисертацията са използвани следните хардуерни модули използвани като ускорители на процеса на обучение, така също и за имплементиране на обучените невронни мрежи.

GPU – Графична карта (GPU) модел GTX 1080 Ti използвана за обучение и тестване на невронни мрежи

Jetson Nano – Използване при имплементиране на обучени невронни мрежи

Movidius - Използване при имплементиране на обучени невронни мрежи

Kendryte K210 - Използване при имплементиране на обучени невронни мрежи

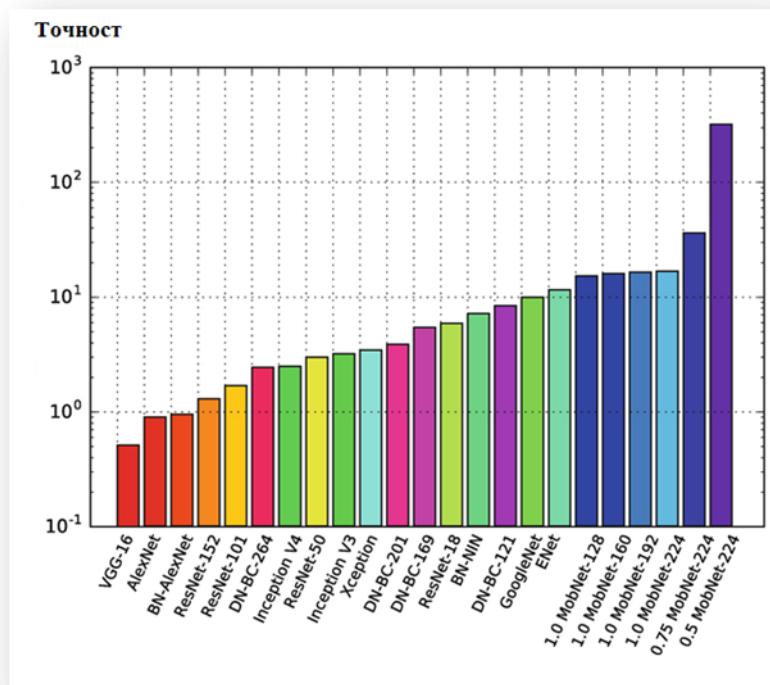
Изводи по глава I

В резултат на направения обзор на алгоритми и приложения могат да бъдат направени следните изводи за изкуствените невронни мрежи:

- Разгледани са различни видове конволюционни невронни мрежи, техни архитектури и алгоритми за обучение.
- Направен е обзор на различни програмни среди позволяващи разработването на невронни мрежи.
- Направен е обзор на различни видове приложения, които използват невронни мрежи.
- Разгледани са различни хардуерни модули за работа с различни видове невронни мрежи.
- Хардуерните ускорители все повече ще навлизат и ще се използват за вградени решения, свързани с невронни мрежи.
- Въз основа на направената литературна справка може да се направи заключението, че използването на изкуствени невронни мрежи за разпознаването на обекти в изображение е динамично развиваща се област в която са инвестират значителни усилия на водещи научни организации.

ГЛАВА II. Разработване на подходи за откриване на агресия на базата на изкуствени невронни мрежи

Разгледана е Pareto графика със съществуващи архитектури на невронна мрежа взимайки основните критерии необходими за задача като откриване на агресия и опасни предмети са избрани невронни мрежи които притежават висока точност и архитектура подходяща за голям набор от устройства [37].



Фиг 2.21 Парето линия при избор на невронна мрежа

От парето графиката са избрани за използвани следните 5 архитектури за сравнение AlexNet, Resnet101, Resnet50, Inception V3, 1.0 MobileNet 224, 0.75 MobileNet 224 като при избор на критерии са взети под внимание портативност, бързодействие и точност.

2.4 Обучение и тестване на невронни мрежа архитектура MobileNet за откриване на опасни предмети

Тренирани са невронни мрежи 1.0 MobileNet 224 и 0.75 MobileNet 224 [142]. Събран е набор от изображения за обекта, за който е необходимо да бъде открит, като изображенията се събират на едно място примерно в папка `positive_images`. В случая са събрани 1000 образа на оръжия, форматиран с фиксиран размер в директорията `positive_images`. По същия начин, са събрани голям набор от отрицателни примери, форматиран подобно на положителните проби и събрани в папка `negative_images`.

Обикновено широкомащабният CNN изисква над един милиард операции за обработка на един вход [149]. CPU с общо предназначение, последователни системи с ограничени изчислителни ресурси, изпълняват твърде неефективно за използване в CNN-базиран изчислителни приложения. Графичните процесори (GPU) широко служат като хардуерни ускорители за обучение на дълбоки невронни мрежи. Въпреки това графичните процесори изискват твърде много енергия за да намерят широко използване във вградени приложения. GPU устройствата могат обаче да се използват за бързо обучение на модела, след което самият модел се имплементира в специализиран хардуер като Movidius или Jetson Nano.

При тестване на работата на невронната мрежа с набор от тестови изображения се получават следните данни :

Таблица 2.4. Процентно съдържание на резултати за откриване на опасни предмети в архитектура MobileNet

Наличие на опасни предмети	Процентно съдържание
True Positive	45.1%
True Negative	43.7%
False Positive	4.9%
False Negative	6.3%

В таблица 2.5 са представени метрики за оценка на работата на MobileNet

Таблица 2.5. Резултати на архитектура MobileNet при мерки за надеждност

Мерка за надеждност	Резултат
Precision	0.848484848
Recall	0.884210526
F-measure	0.865979381

Въз основа на направеното допълнително обучение на MobileNet 224 става възможно откриването на опасни предмети във изображение.



Фиг 2.28 Резултати при търсено оръжие и точност при обучение MobileNet невронна мрежа с TensorFlow

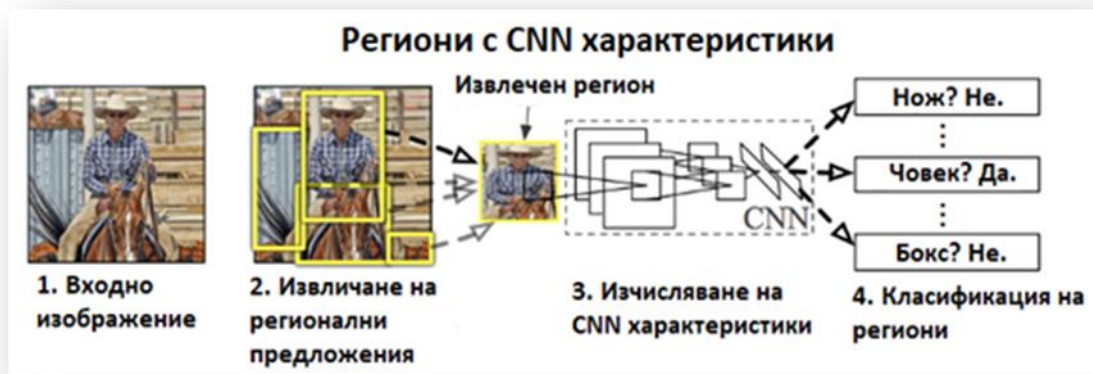
2.5 Обучение на R-CNN Inception и Resnet използвайки TensorFlow

При тренирането са използвани същата база от данни като за преходните архитектури. За създаването на собствен модел за разпознаване на желани обекти първо е необходимо да бъдат събрани изображения с търсените обекти сортирани в папка [143]. Преди обучението е необходимо всички изображения да бъдат с еднаква резолюция, това е реализирано със скрипт на програмен език Python който се изисква със следната команда: `python transform_image_resolution.py -d images/ -s 800 600`.

Предложения код взема като входен параметър всички изображения в директория "images" и ги форматира с резолюция 800 x 600 пиксела след което със специализиран софтуер се извличат параметрите на търсения обект от изображението като е желателно то да не съдържа излишни обекти. Всяка невронна мрежа поддържа изображения с различна резолюция поради тази причина е необходимо входните данни да бъдат съвместими с нея, в противен случай точността ще бъде значително по-ниска. След като характеристиките на обектите са извлечени за обучението на невронната мрежа, TensorFlow изисква входния

параметър да бъде във формат “ tfrecord “, който може да бъде създаден с командата: python generate_tfrecord.py .

Предложените архитектури използват селективно търсене за извличане на региони от изображението, наречени регионални предложения. По този начин, вместо класифицирането на голям брой региони, се работи с голям брой региони. Тези регионални предложения се генерират чрез алгоритъм за селективно търсене [3].



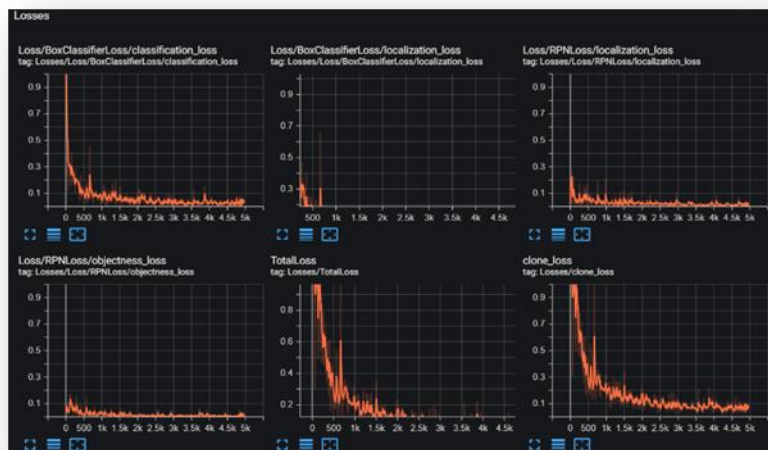
Фиг 2.29 Принципно разпознаване и класификация с R-CNN Inception и Resnet

Със получения tfrecord файл невронната мрежа може да бъде обучение създавайки модел с възможност за имплементиране на множество места за разпознаване и класификация на търсените обекти. Мониторирането на обучението може да бъде видно в cmd (Command Prompt). То включва безкраен цикъл, който показва броя итерации при всяка стъпка. Освен това за по-голяма яснота обучението може да бъде наблюдавано в “TensorBoard “ и представено в графичен вид [93].

```
INFO:tensorflow:global step 4902: loss = 0.1191 (9.692 sec/step)
INFO:tensorflow:global step 4903: loss = 0.1005 (10.344 sec/step)
INFO:tensorflow:global step 4903: loss = 0.1005 (10.344 sec/step)
INFO:tensorflow:global step 4904: loss = 0.0622 (9.361 sec/step)
INFO:tensorflow:global step 4904: loss = 0.0622 (9.361 sec/step)
```

Фиг 2.30 Обучение на R-CNN Inception в TensorFlow

Обучението обикновено започва с висока стойност на параметъра Loss (или грешка при обучението). В процеса на обучение стойността на параметъра става по-ниска и по-ниска.



Фиг 2.32 Обучение на невронни мрежи в TensorFlow – промяна на параметъра Loss

Когато обучението е успешно, или параметъра Loss намалее до стойност под 0.1, може да се приеме, че невронната мрежа е обучена с достатъчно добра степен на точност за разпознаване на търсените обекти в изображение. Ниската стойност на параметъра за грешка се изразява във висока вероятност при наличие на търсения обект, той да бъде разпознат правилно и с висока степен на достоверност.

При тестване на работата на невронната мрежа с набор от тестови изображения се получават следните данни :

Таблица 2.6. Процентно съдържание на резултати за откриване на опасни предмети в архитектура R-CNN

Наличие на опасни предмети	Процентно съдържание
True Positive	42.6%
True Negative	42.45%
False Positive	7.4%
False Negative	7.55%

В таблица 2.7 са представени метрики за оценка на работата на R-CNN

Таблица 2.7. Резултати на архитектура R-CNN при мерки за надеждност

Мерка за надеждност	Резултат
Precision	0.831683168
Recall	0.875
F-measure	0.852791878

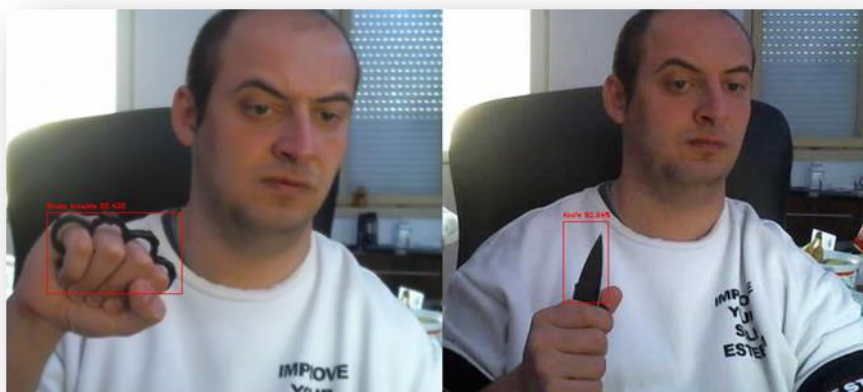


Фиг 2.33 Резултат при търсено оръжие и точност при обучение на R-CNN невронна мрежа с TensorFlow

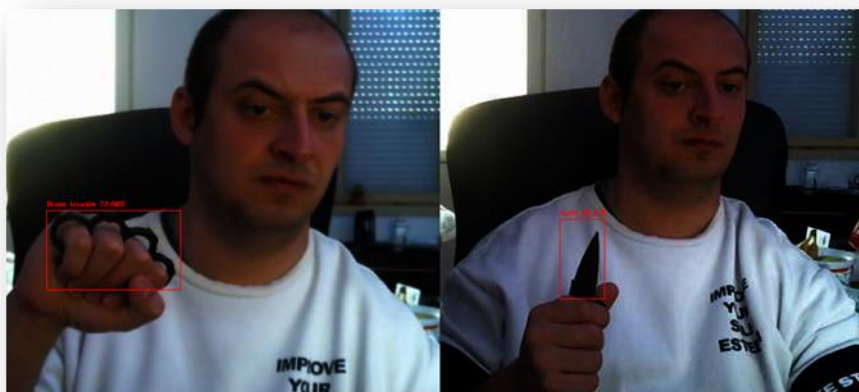
2.6 Разработване на алгоритъм за адаптивно разпознаване на обекти с динамична размяна на невронни мрежи

За да се приеме, че един алгоритъм за разпознаване на обекти в изображения работи успешно, то той трябва да връща коректни резултати при различни изменения в яркостта и контраста на входните изображения.

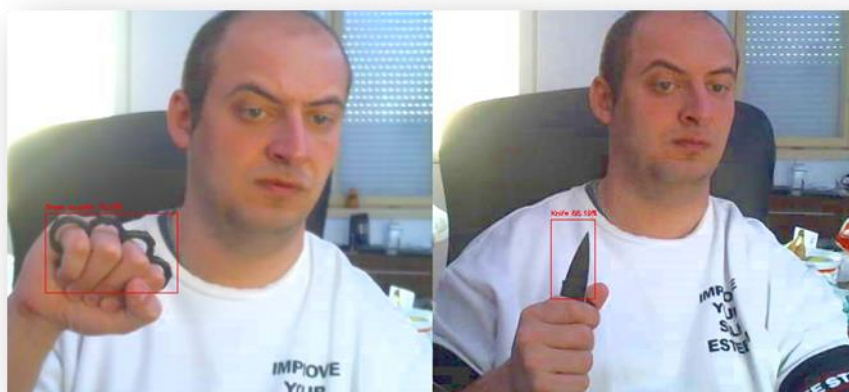
Невронните мрежи, обучени за разпознаване на опасни предмети, са тествани при различна яркост използвайки стандартна USB камера и тестови софтуер разработен на програмен език Python. Показани са резултатите при различна яркост като е включен брояч показващ обработените брой кадри.



Фиг 2.30 Открити обекти при оригинална картина

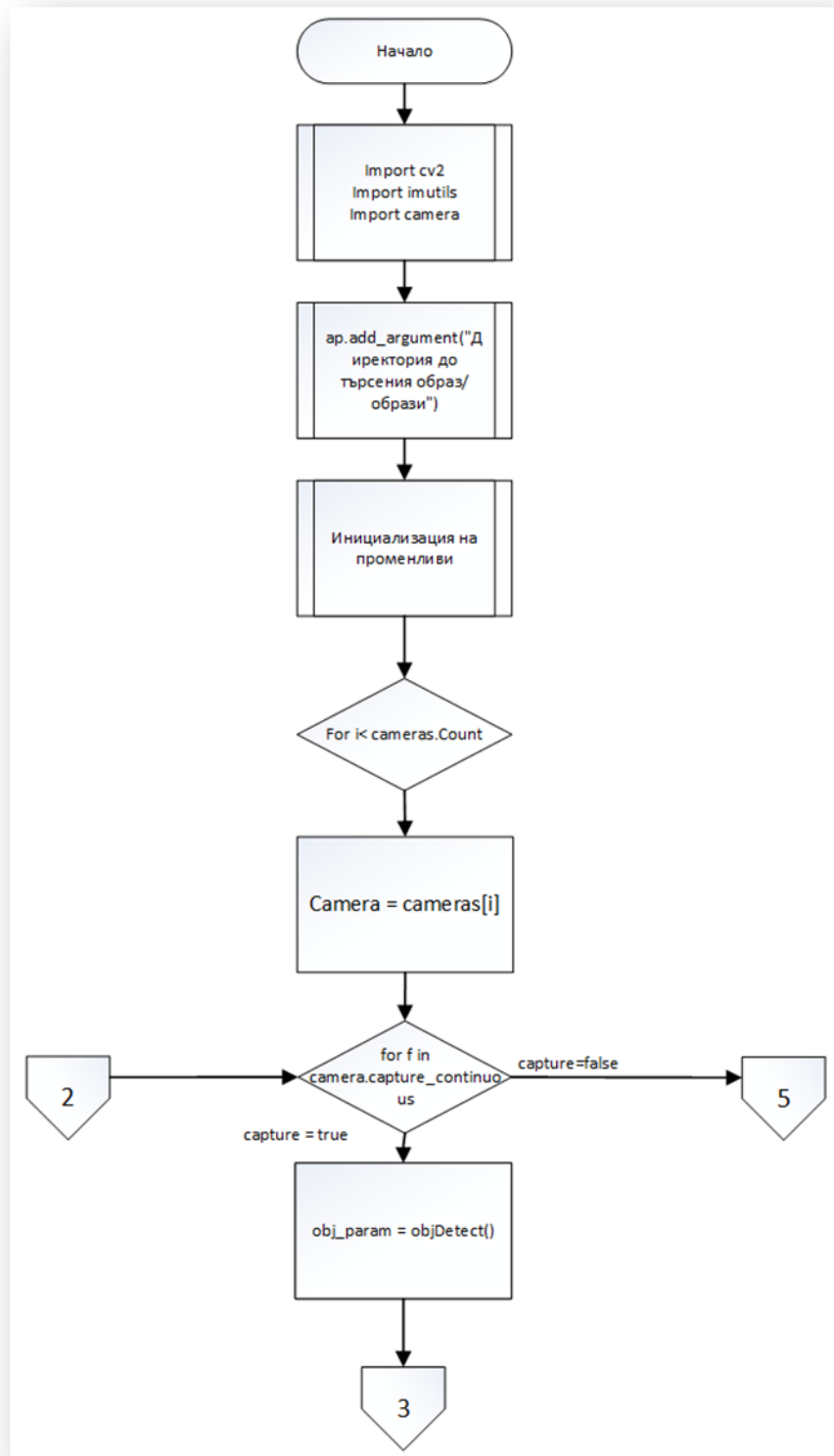


Фиг 2.31 Открити обекти при затъмненост 50%



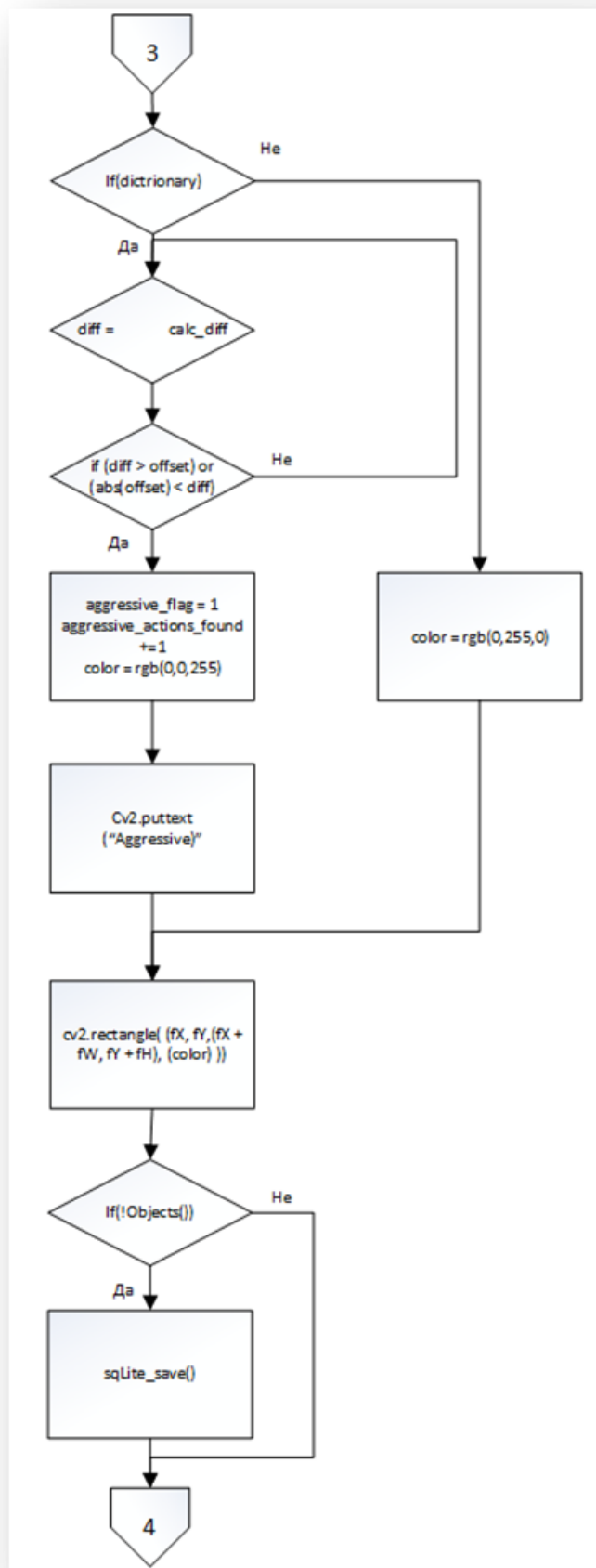
Фиг 2.32 Открити обекти при осветеност 50%

Алгоритъмът позволява да се търси и открива в изображение опасен предмет с предоставянето на необходимата класификационна информация за него. Той може да бъде използван върху различни процесори, което дава възможност на кода да бъде разполаган върху много видове платформи и съответно така да се тества неговото бързодействие. На места като стадиони, площади или други публични места, където има голям брой от хора и голяма вероятност за проява на опасни предмети, човек не може да отреагира достатъчно бързо и поради тази причина човешкия фактор може да се замени със система, която ако открие наличието на опасен предмет в изображението, автоматично да сигнализира за това.. Възможно е множество камери от различни места по IP да предоставят видео информацията към централен компютър, в който работи софтуера за разпознаване и класификация. За това е нужен сравнително по-мощен процесор, но за сметка на това всички операции ще бъдат извършвани от едно място. Примерен алгоритъм за разпознаване на обекти от множество камери е показан на фигура 2.37.



Фиг 2.37 а) Примерен алгоритъм за откриване на обекти от множество камери

На първата част от алгоритъма е показана инициализацията на системата и зареждане на камерите, зареждането на невронната мрежа и цикъл, в който камерите биват обхождани за събиране на информация от тях.



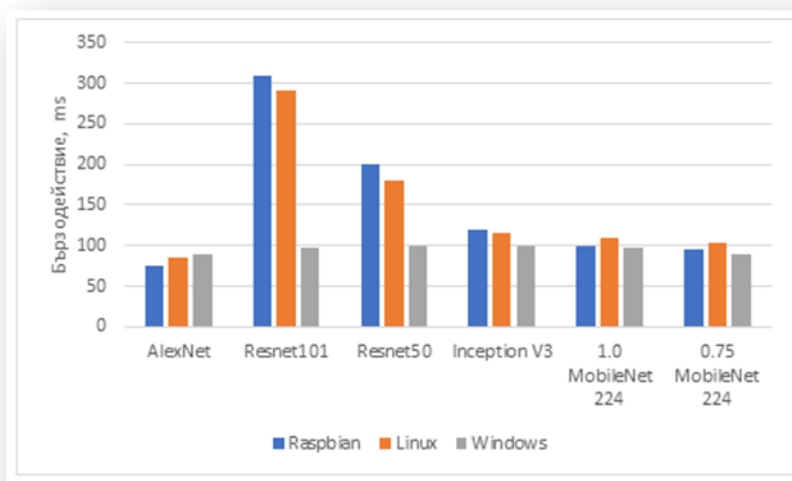
Фиг 2.37 б) Примерен алгоритъм за откриване на обекти от множество камери

Работата на алгоритъма на базата на MobileNet 224 е сравнен с работата на други модели на конволюционни невронни мрежи. Предложените архитектури са тествани върху платформите Raspbian, Linux и Windows като тяхното бързодействие е показано в таблица.

Таблица 2.8. Бързодействие (в msec) при различни CNN архитектури

ОС	Бързодействие на обработката, msec					
	AlexNet	Resnet101	Resnet50	Inception V3	1.0 MobileNet 224	0.75 MobileNet 224
Raspbian	75	310	200	120	100	95
Linux	85	290	180	115	110	103
Windows	90	97	100	100	97	90

Както се вижда от таблицата най-голямо бързодействие има AlexNet, но трябва да се вземе под внимание, че тя е невронна мрежа за класификация, а не за локализиране на обекти за разлика от MobileNet, Inception и Resnet. Направено е сравнение между тяхната работа като тяхната точност и бързодействие са тествани с 1000 кадъра от видео използвайки предложения алгоритъм. Резултатите от направените изследвания за точност са визуализирани във вид на графика на фиг 2.38.



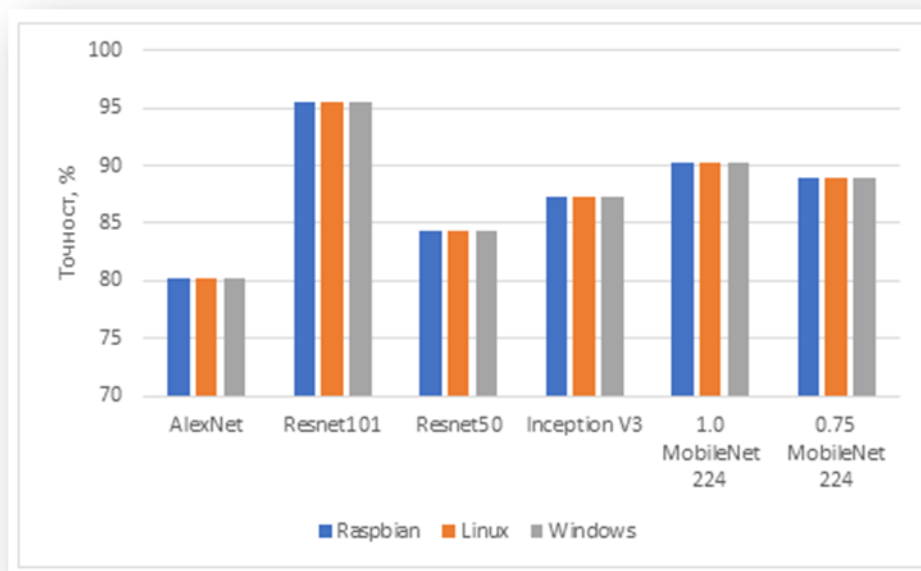
Фиг 2.38 Графика показваща бързодействие на различни архитектури

Също така е направено сравнение за точност върху различните операционни системи, както се вижда от таблица 2.9 и фигура 2.35, MobileNet и Resnet101 имат по-висока точност като операционната система не оказва въздействие върху тях.

Таблица 2.9. Точност при различни CNN архитектури в нормална осветеност

ОС	Точност при обработка, %					
	AlexNet	Resnet101	Resnet50	Inception V3	1.0 MobileNet 224	0.75 MobileNet 224
Raspbian	80.2	95.5	84.4	87.3	90.2	88.9
Linux	80.2	95.5	84.4	87.3	90.2	88.9
Windows	80.2	95.5	84.4	87.3	90.2	88.9

На фигура 2.39 е направена визуализация на направените изследвания при нормална осветеност.



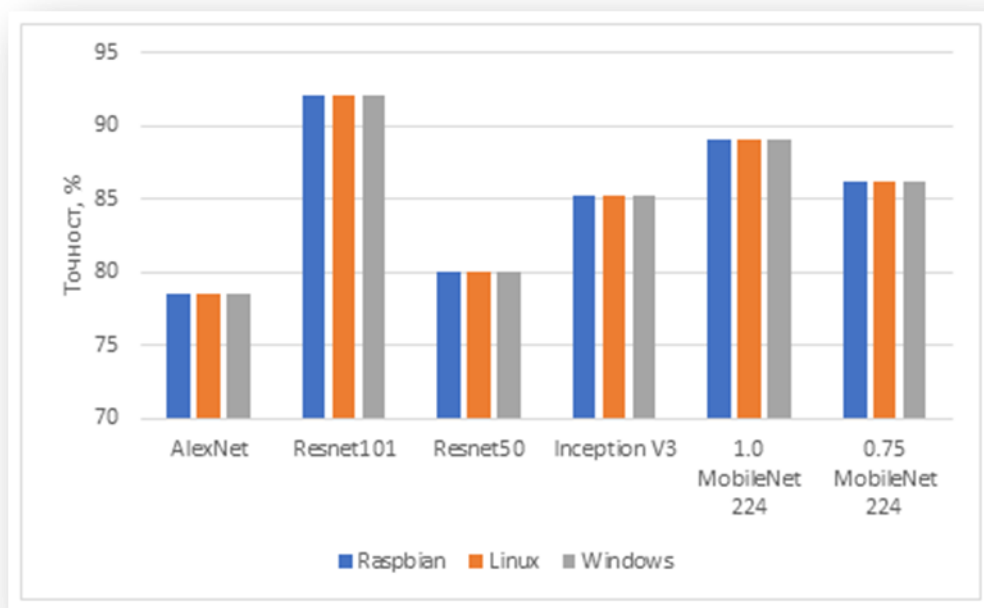
Фиг 2.39 Графика показваща точност на различни архитектури при нормална осветеност

При различна осветеност точността на невронните мрежи може да варира и са направени тестове за точност при осветеност плюс и минус 50%. Резултатите от тези изследвания са показани в таблици 2.10 и 2.5

Таблица 2.10. Точност при различни CNN архитектури при осветеност увеличена със 50%

ОС	Точност при обработка, %					
	AlexNet	Resnet101	Resnet50	Inception V3	1.0 MobileNet 224	0.75 MobileNet 224
Raspbian	78.5	92.1	80.1	85.2	89	86.2
Linux	78.5	92.1	80.1	85.2	89	86.2
Windows	78.5	92.1	80.1	85.2	89	86.2

Резултатите от направените изследвания при осветеност със 50% са визуализирани на фигура 2.40

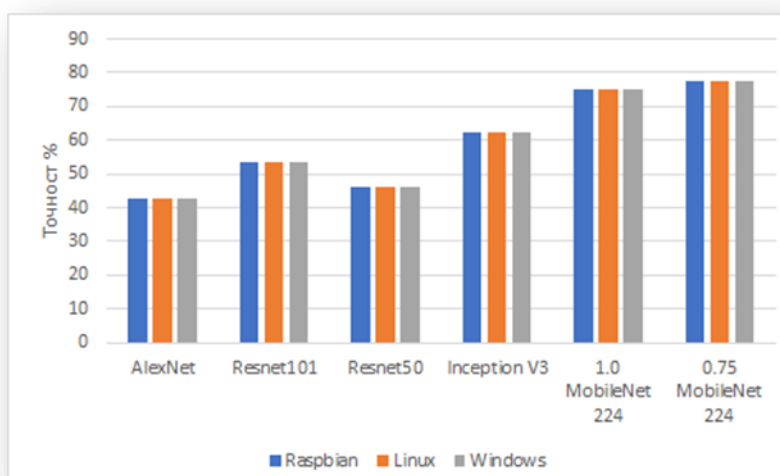


Фиг 2.40 Графика показваща точност на различни архитектури при осветеност увеличена със 50%

Таблица 2.11 Графика показваща точност на различни архитектури при осветеност намалена със 50%

ОС	Точност при обработка, %					
	AlexNet	Resnet101	Resnet50	Inception V3	1.0 MobileNet 224	0.75 MobileNet 224
Raspbian	42.5	53.4	46.1	62.3	75.1	77.4
Linux	42.5	53.4	46.1	62.3	75.1	77.4
Windows	42.5	53.4	46.1	62.3	75.1	77.4

От изследванията при ниска осветеност се вижда, че архитектурата Resnet101 има значително по-ниска точност. На фигура 2.41 са показани визуалните резултати.



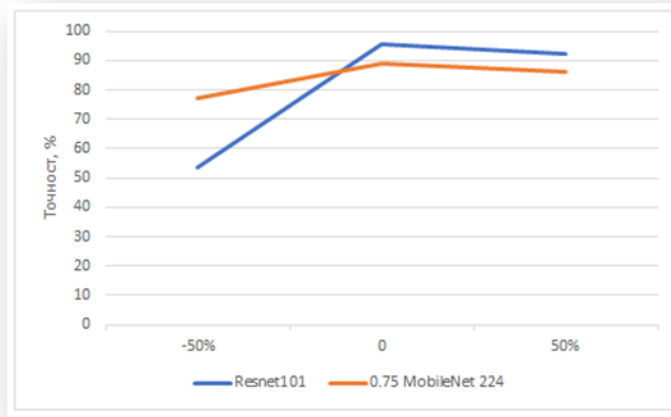
Фиг 2.41 Графика показваща точност на различни архитектури при осветеност намалена със 50%

Както се вижда при ниска осветеност MobileNet се представя по-добре от Resnet101 сравнението между тях е показано в таблица 2.6

Таблица 2.12 Сравнение между Resnet101 и 0.75 MobileNet при различна осветеност.

Осветеност	Resnet101	0.75 MobileNet 224
-50%	53.4	77.4
0	95.5	88.9
50%	92.1	86.2

Данните са визуализирани на фигура 2.42



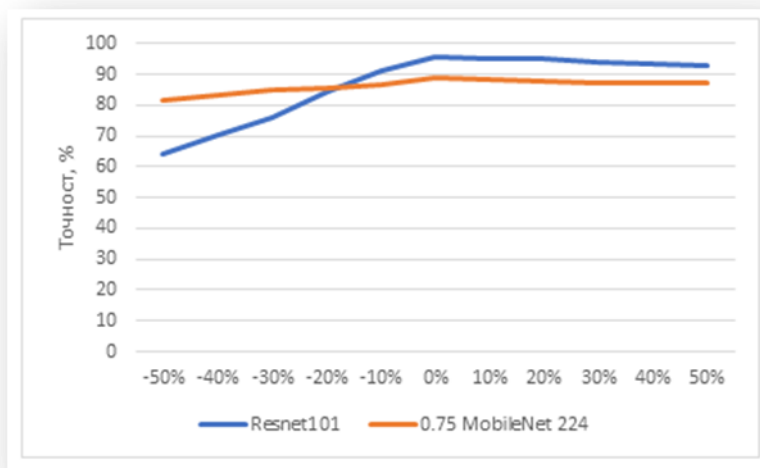
Фигура 2.42 Сравнение между Resnet101 и 0.75 MobileNet при различна осветеност.

Разработен е код на Python който променя яркостта със стъпка 10% и е направено проучване как се променят точностите между двете архитектури при различна осветеност. Резултатите са показани в таблица 2.7.

Таблица 2.13 Сравнение между различните архитектури за открити резултати със стъпка 10%

Осветеност	Resnet101	0.75 MobileNet 224
-50%	64.3	81.3
-40%	70.1	83.2
-30%	75.7	84.7
-20%	84.6	85.2
-10%	90.9	86.7
0%	95.5	88.9
10%	95.3	88.5
20%	94.8	87.9
30%	94	87.4
40%	93.5	87.1
50%	92.7	87

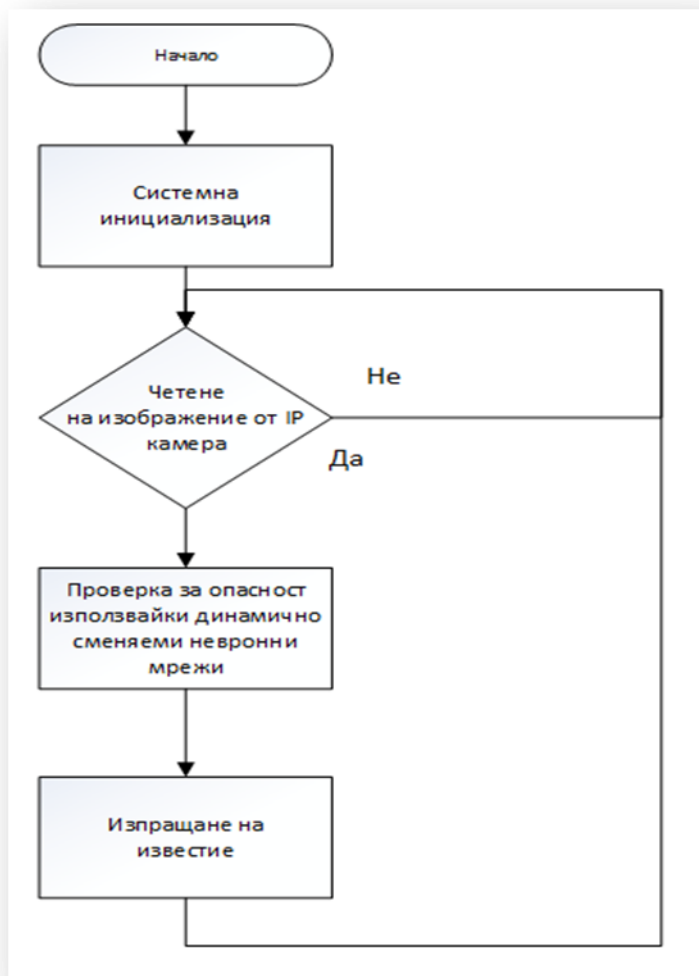
От таблицата се вижда, че при около – 20% двете архитектури имат сравнително еднаква точност и при намаляване на яркостта MobileNet има по-добри резултати. Изследването е показано в графичен вид на фигура 2.43.



Фигура 2.43 Сравнение между Resnet101 и 0.75 MobileNet при промяна на осветеността със стъпка от 10%.

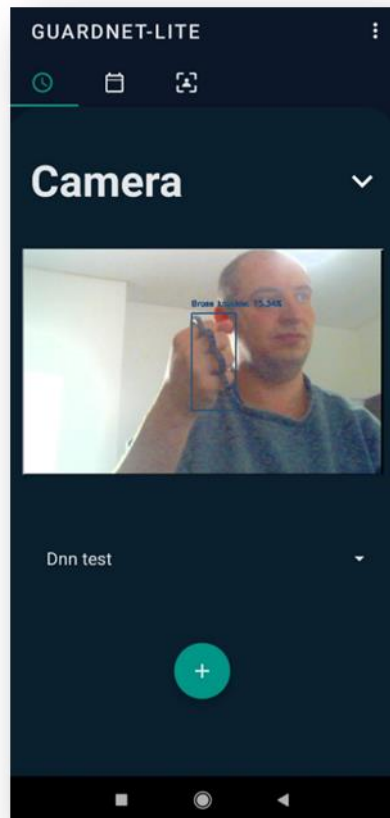
Разработване на мобилно приложение за откриване на оръжие и опасни предмети с използването на избраните невронни мрежи

Като вследствие на направените изследвания и предложения алгоритъм за разпознаване на опасни обекти при различна осветеност е разработено мобилно приложение което използва публична или лична IP камера. Предлага се използването на изследваните невронни мрежи Resnet101 и MobileNet. Алгоритъмът на разработената система е показан на фигура 2.44. Той използва 2 архитектури на невронни мрежи, които се сменят динамично при различна яркост която се определя със стандартни функции от библиотеката OpenCV



Фиг 2.44 Алгоритъм на работа

Началото на алгоритъмът се състои от инициализация на система и извличане на изображения от камерата след което се проверява за агресия и опасни предмети. Проверката за опасност открива различни видове оръжия използвайки избраните невронни мрежи както и човешкото позициониране. При открита опасност се изпраща нотификация. Той е интегриран в мобилно приложение за Android операционна система която взема информация от публична уеб камера на град Габрово или от лична камера от такъв тип. Приложението съдържа списък с камерите показани на фигура 2.45 като има конфигурация за нотификация при каква продължителност на насилие да алармира съответните органи



Фиг 2.47 Открито оръжие в мобилно приложение

Направеното мобилно приложение представлява завършек на направените изследвания притежавайки потенциал да се използва в различни среди

Изводи по Глава 2

В резултат на изследваните алгоритми и тяхното приложение в различни среди могат да бъдат направени следните изводи за разпознаване и класификация:

- Разгледани са различни алгоритми за откриване на агресия.
- От Pareto графика са избрани невронни мрежи с подходящата архитектура за настоящите изследвания и избраните невронни мрежи са тренирани с подходяща база данни.
- В главата е направено изследване на различни модели DNN за разпознаване на обекти в изображение като работата на моделите е тествана при различна осветеност.
- Въз основа на направените експериментални изследвания е предложен алгоритъм за адаптивно търсене на опасни предмети в изображения с промяна на търсеция DNN модел в зависимост от осветеността.
- Разработено е мобилно приложение за откриване на оръжие и опасни предмети.

ГЛАВА III. Приложение на хардуерни ускорители за невронни мрежи за разпознаване на жестове

3.1 Разработване на система за откриване и изпълняване на команди от жестове открити над главата на човешки обект

Разпознаването на жестове е подход с приложение в множество области. Едно от тези приложения е при индустриалните системи, когато с помощта на жестове ще могат да се спират, забавят или ускоряват машини или процеси с цел да се постигне висока сигурност на производството. Цел на изследванията е разпознаване на жестове с помощта на изкуствена невронна мрежа.

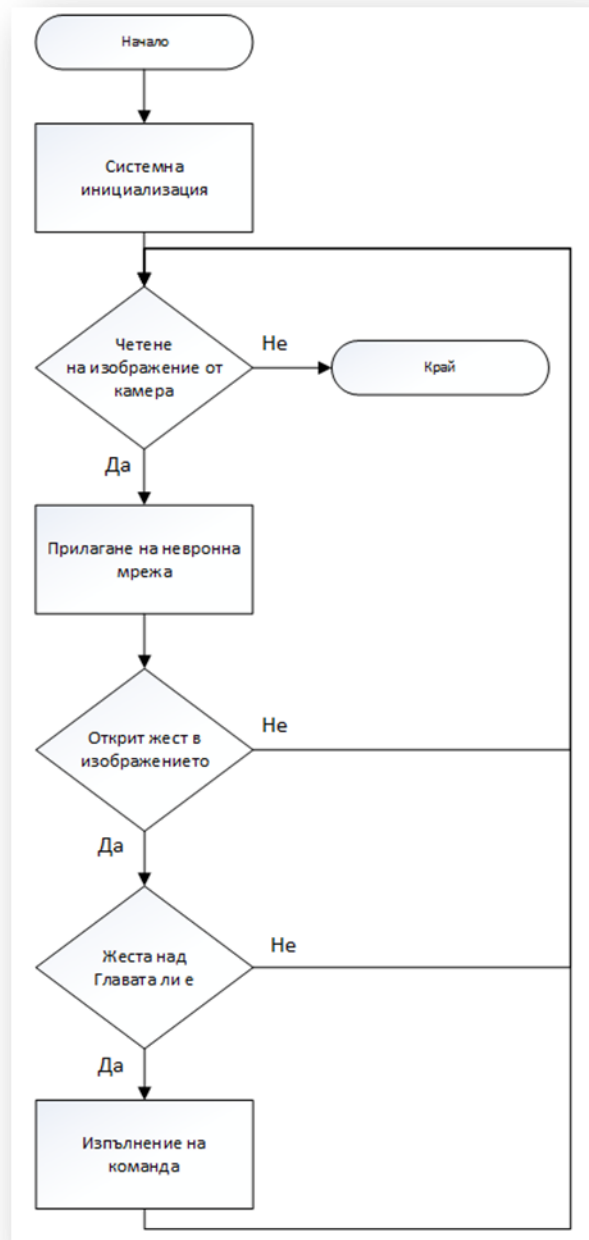
Обучена е невронна мрежа с MobileNet архитектура която осигурява висока скорост при откриване на жестове и портативност между различни платформи. Обучението изисква предварителна база данни със снимки, от които е необходимо да се извлекат характеристиките на жестовете, които трябва да бъдат намерени след това.

Фиг 3.11 показва части от снимките на 2 жеста в различни изгледи като за тренирането на невронна мрежа са използвани по 200 снимки на жест с общ брой от 5 жеста.



Фиг. 3.11 Част от снимки използвани за трениране на невронна мрежа за откриване на жестове

За обучението се използва библиотеката за машинно обучение TensorFlow и език за програмиране Python, Невронната мрежа е в състояние да разпознава множество жестове и чрез прилагане на софтуерна филтрация кой обект е жест, човешка глава или тяло е възможно да се обработват команди въз основа на данните от изображението. Алгоритъмът за изпълняване на жестове открити над човешката глава е представен на фигура 3.12



Фиг. 3.12 Алгоритъм за откриване на жестове в изображение

След прочитането на всеки кадър той се сканира, като се използва обучената изкуствена невронна мрежа за разпознаване на жестове. Ако жестът бъде намерен и той е над главата на човека, се изпълнява команда. Предложен е алгоритъм за валидация на жестове като за валиден жест е необходимо той да бъде разположен над човешката глава, а в случай, че той бъде под нея командата се смята за невалидна.

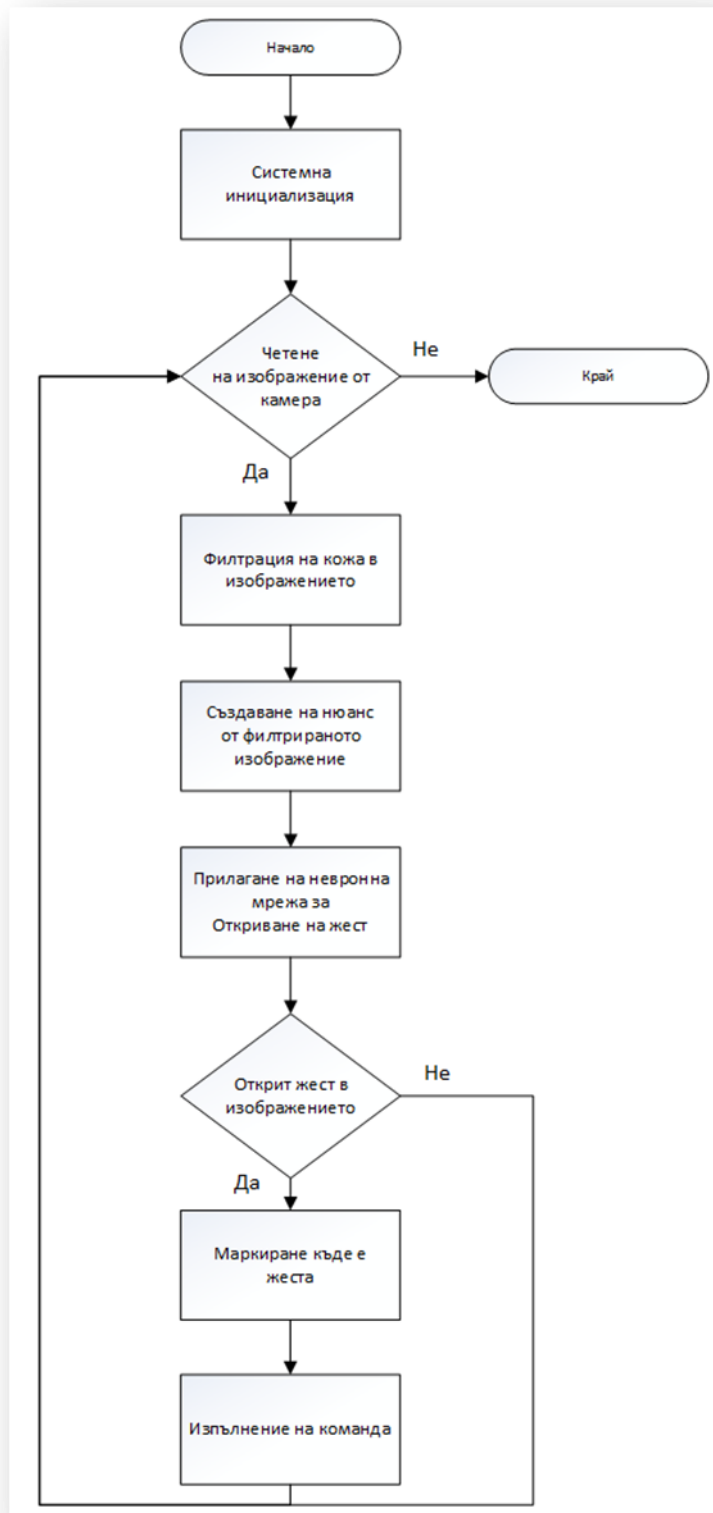


Фиг. 3.13 Валидно / невалидно състояние на команда 1



Фиг. 3.14 Валидно / невалидно състояние на команда 2

Системата има висока точност, но при голям набор от жестове невронната мрежа може да върне грешен жест поради което е необходимо допълнителна филтрация. За тази цел се предлага използването на софтуерна филтрация за откриване на човешка кожа в изображение и извличане на нейният нюанс. Предложеният алгоритъм е представен на фигура 3.16.



Фиг. 3.15 Алгоритъм за откриване на жестове във видео поток

Системата работи като първо извлича човешката кожа, след което търси силует на жест, който е трениран от невронна мрежа. При открит силует се изчертава неговата локация върху реалното изображение.

Адаптивното извличане става чрез няколко проверки за цвета на кожата като на този етап системата поддържа само бялата раса с възможност за разширение за други раси (жълти, кафяви). При успешно извличане ръката се слага на черен фон при което се взима нейният

нюанс. Това е необходимо понеже различни хора нямат идентични ръце и този процес уеднаквява човешката ръка. Полученият резултат се предава на невронна мрежа която връща номер на жест и неговата локация.



Фиг. 3.16 Откриване на жест 1



Фиг. 3.17 Откриване на жест 2



Фиг. 3.18 Откриване на жест 3

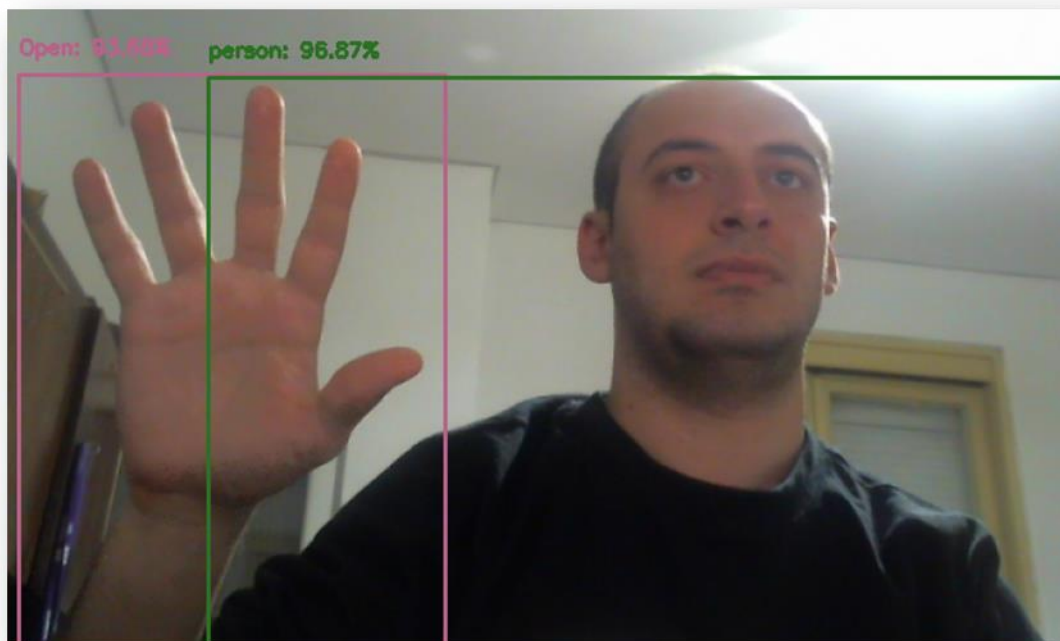


Фиг. 3.19 Откриване на жест 4

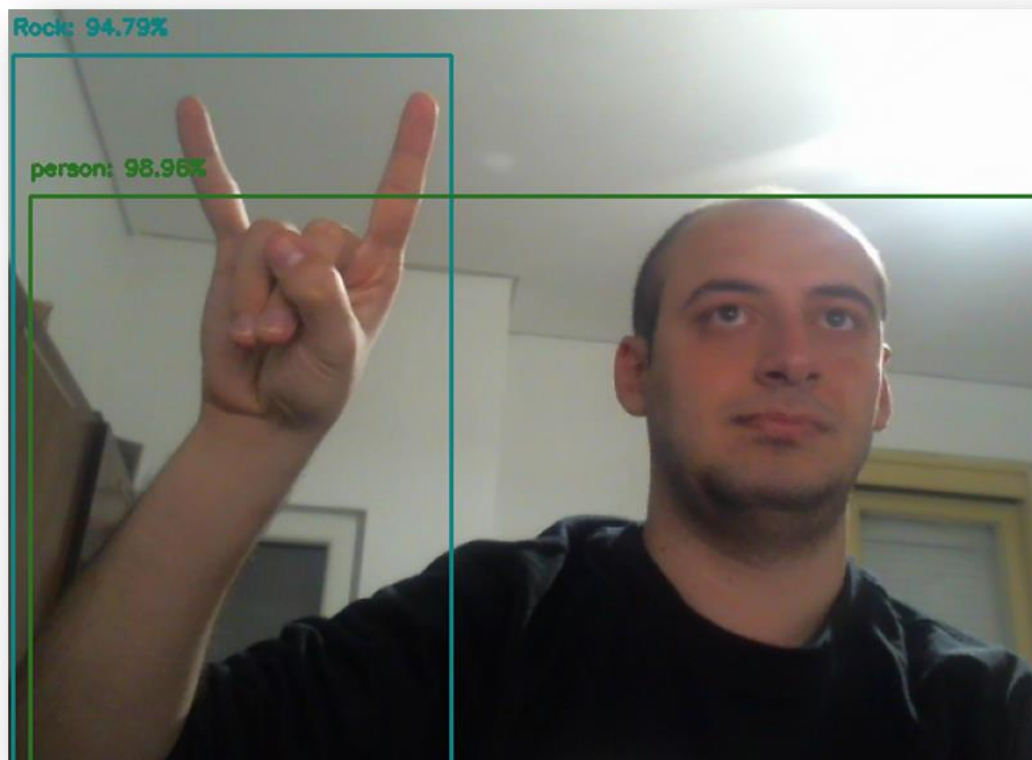


Фиг. 3.20 Откриване на жест 5

След откриването на жеста неговата локация както и локацията на човешкият обект се прилага върху оригиналното изображение. Крайният резултат е показан на фигура 3.21



Фиг. 3.21 а) Открита команда 1 от човешки обект



Фиг. 3.22 б) Открита команда 5 от човешки обект

С предложеният алгоритъм системата е тествана с 5 на брой жестове и може да работи надеждно с много висока точност и да бъде интегрирана в автоматизацията на различни процеси. Тя има потенциал да бъде доразвита като жестовете софтуерно могат да се адаптират използвайки идентификатор, кой жест от кой човек е.

Подобни видове системи могат да бъдат използвани в индустрията за автоматизация и управление на различни видове работи.

3.2 Откриване на дистанция в реално време предназначена за безопасна среда при работа на индустриален робот в човешко присъствие

Човешката дистанция при работа на машина е важен фактор за неговата безопасност и при прекалена близост би могло да окаже фатален край. За да се избегнат подобни инциденти в дисертацията се предлага система за откриване на дистанцията между човек и индустриален робот, като при прекалена близост работата на робота се спира. За разработването на такава система ключов фактор е калибровката, настройката и бързодействието на системата за бързи и коректни реакции [107],[108]. Калибрирането е реализирано знаейки ориентировъчната широчина на човека, неговата ширина в пиксели на фиксирано разстояние и фокусът на камерата и може да се изчисли по следната формула [109]:

$$f = d \cdot l / D \quad (3.1)$$

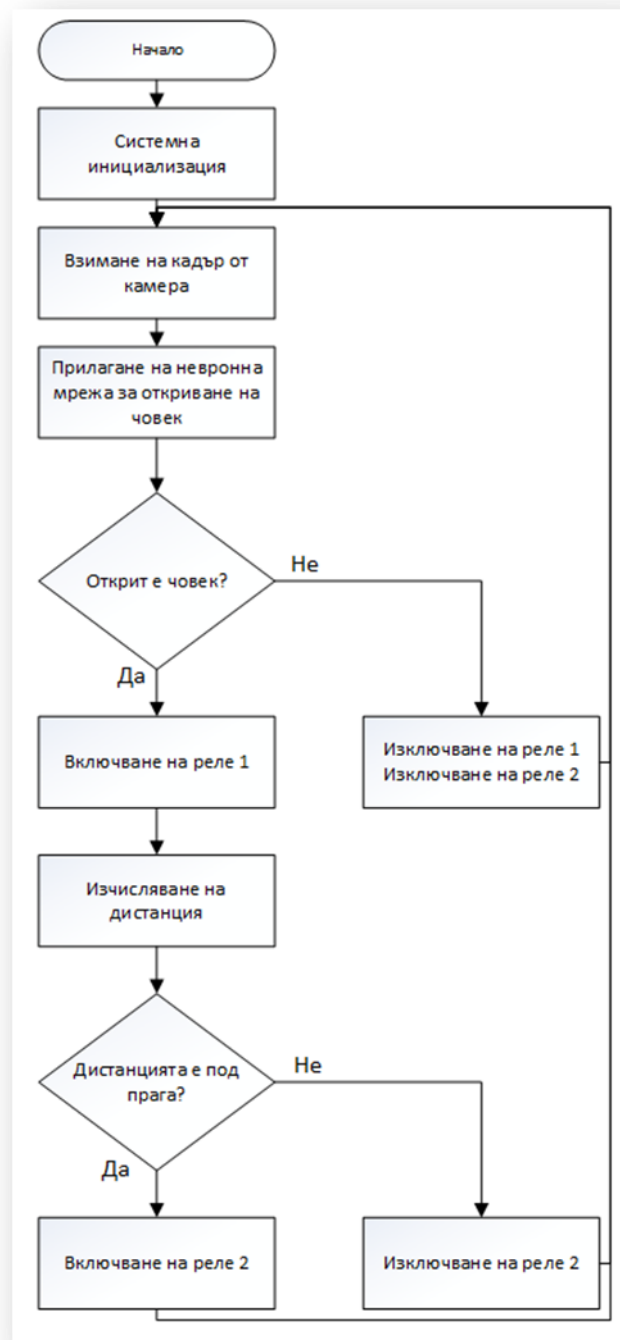
където f - фокус на камерата, D – Ширина на обекта в сантиметри, d – Ширина на обекта в пиксели, l – Разстояние от обекта

Знаейки тези параметри следователно разстоянието може да бъде открито по следната формула

$$l = D \cdot f / d \quad (3.2)$$

Началото на алгоритъма започна с импортирането на библиотеките CV2, Imutils и Camera и добавяне на аргумент съдържащ обучената невронна мрежа за търсеня обект.

Понеже за такъв вид системи бързодействието е ключов фактор за бързото реагиране е използван Movidius USB Stick. При работата се проверява комуникацията с него и при неуспешна комуникация програмата се прекратява. След успешна инициализация управляващият сигнал за човешко присъствие е неактивен, а сигнала за активна работа активиран. Прави се инициализация на камерата и се задава желаната резолюция. При успешно отваряне на видео камерата започва цикъл, в който всеки получен кадър се сканира за откриване на човек. При откриването на такъв се задейства управляващ сигнал, че е открито човешко присъствие. Ако той е в опасна близост до работещата машина, то тя се спира, до като обекта не се е отдалечил на безопасно разстояние от индустриалния робот. Алгоритъмът на програмата е представен на фиг 3.23.



Фиг. 3.23 Предложен алгоритъм за откриване на дистанция на човешки обект и неговата безопасност с помощта на машинно зрение

Както става ясно от фигура 3.24 софтуерът започва със инициализация на периферните устройства и при успех се извличат входни изображения в реално време. Към системата за определяне на дистанция са включени две релета – реле 1 за човешко присъствие и реле 2 за прекалена близост. При успешно извлечено изображение то бива сканирано за обекти, чиито параметри са зададени в предварително обучена невронна мрежа. Ако е намерен човек, реле 1 се активира и ако той е на повече от 80 см, реле 2 се изключва. Съответно при неговото отдалечаване или неговото отсъствие от камерата реле 2 подава управляващ сигнал на индустриалния робот да продължи своята работа както и реле 1 се изключва до като не бъде открит нов обект.

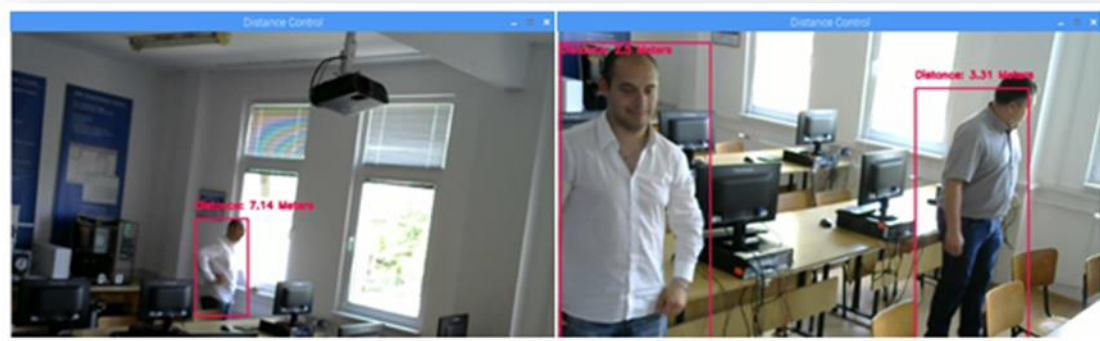
Околната среда около робота може да се класифицира в три категории - зоната на безопасност, зоната за тревога и опасна зона (фиг.3.24). Зоната за предупреждение е извън обсега на робота, но когато има човешко присъствие, системата на камерата трябва да я открие и да изпрати съответния сигнал към робота.



Фиг. 3.24 Зона около робот

Когато човешки обект е в зоната за тревога или опасност, системата включва второто реле правейки робота в неактивно състояние до момента, в който обекта не премине в зоната на безопасност.

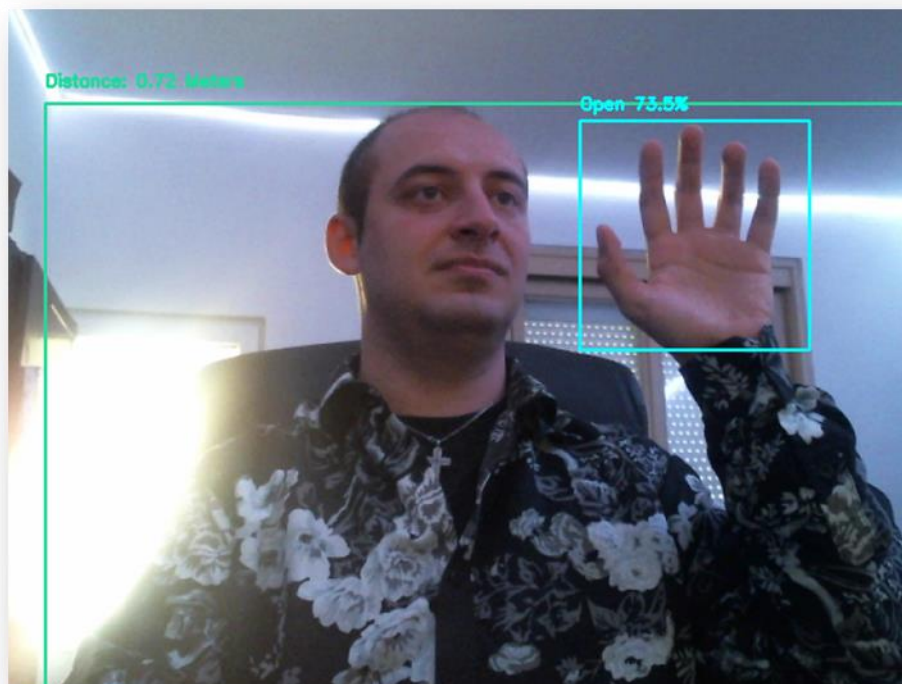
При успешно свързване на предложената система към индустриалния робот е поставена камера, от която могат да се видят следните резултати показани на фиг 3.26. Както се вижда от нея, до откритите хора има индикатор за тяхната дистанция. Системата работи с еднакво бързодействие независимо от броя на откритите обекти.



Фиг 3.25 Откриване на дистанция до човек с използване на машинно зрение.

Системата за откриване на дистанция може да бъде използвана в реализирането на отдалечено управление на работи, откривайки безопасното разстояние до човек преди изпълнението на команда. По този начин командата ще бъде изпълнена само ако човекът се намира извън зоната непосредствено до робота.

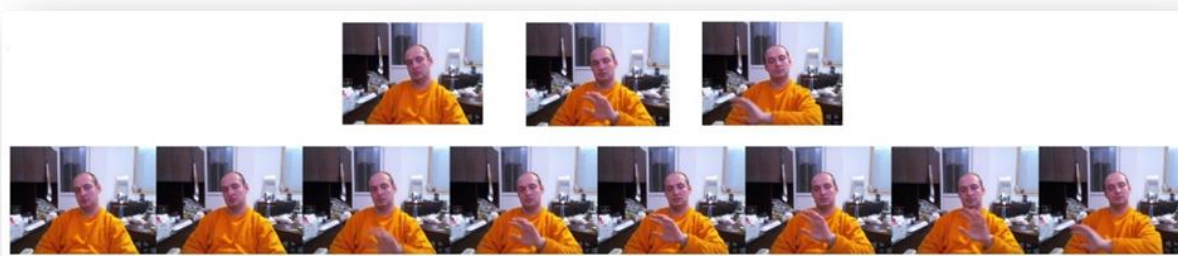
В комбинация с предложената система за жестове, системата е възможно да бъде използвана не само в индустриални помещения за управление на различни видове индустриални работи, а също и при управлението на сервизни работи в домакинството или за включване на определени функции от т.нар. интелигентен дом.



Фиг 3.26 Откриване на човешка дистанция с използване на машинно зрение.

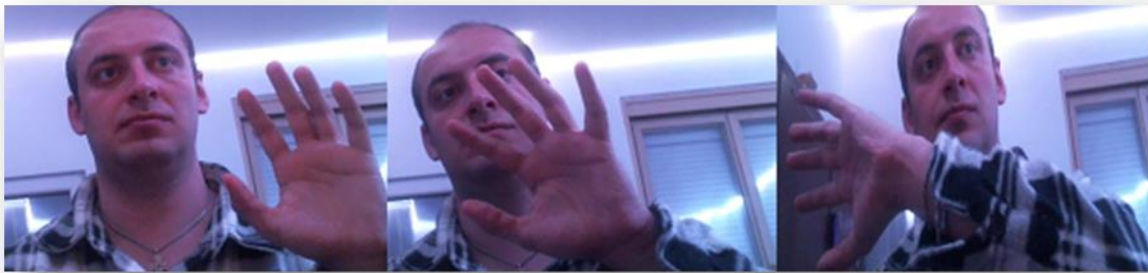
3.3 Обучение на мрежата върху кратки видео последователности

Откриването на действия е възможно чрез анализ на серия от изображения направени във времето. За тази цел се използва подхода Backpropagation Through Time предоставен от Keras позволяващ лесното и проектиране. За обучението на невронната мрежа са необходими видео кадри с дължина около 1 секунда.



Фиг 3.41 Извличане на данни от видео поток

За обучението на невронната мрежа се използват кратки видео последователности в продължителност от около 1 секунда със замах на ръка показани на фигура 3.42



Фиг. 3.42 Кадри от видео за откриване на ляв замах с ръка



Фиг. 3.43 Кадри от видео за откриване на десен замах с ръка

Keras може да създава големи LSTM мрежи, които от своя страна могат да бъдат използвани за справяне с трудни проблеми при машинно обучение и постигане на най-високи резултати. Вместо неврони тези мрежи притежава блокове памет, които са свързани между слоевете. Блокът има компоненти, които го правят по-интелигентен от невронът и памет за скорошни последователности. Блокът съдържа гейтове, които управляват състоянието и изхода на блока. Той работи върху входна последователност и всеки гейт в рамките на блока използва сигмоидни активиращи функции, за да контролира дали те са задействани или не, като прави промяна на състоянието и добавяне на информация, преминаваща през блока.

Има три вида гейта в рамките на един блок:

1. Гейт за забравя: условно решава каква информация да изхвърлите от блока;
2. Входен гейт: условно решава кои стойности от входа да актуализират състоянието на паметта;
3. Изходен гейт: условно решава какво да се изведе въз основа на входа и паметта на блока.

Всеки блок работи като машина на състоянието на която гейтовете имат тежести, които се научават по време на тренировъчен процес. Ефективно обучение се получава когато данните се преразпределят в диапазона от 0 до 1, това действие се нарича още нормализиране. Най-лесният начин за извършване на тази обработка е чрез използването на класа за предварителна обработка `MinMaxScaler` от библиотеката `scikit-learn`. След моделирането на данните и установяването на точността, използването на базата данни за обучение, е необходимо да се направи проверка с данни, които не са били включени в обучението. Мрежата има видим слой с 1 вход, скрит слой с 4 LSTM блока и един изходен слой прогнозиращ. По подразбиране се използва сигмоидна функция за активиране на LSTM блоковете.

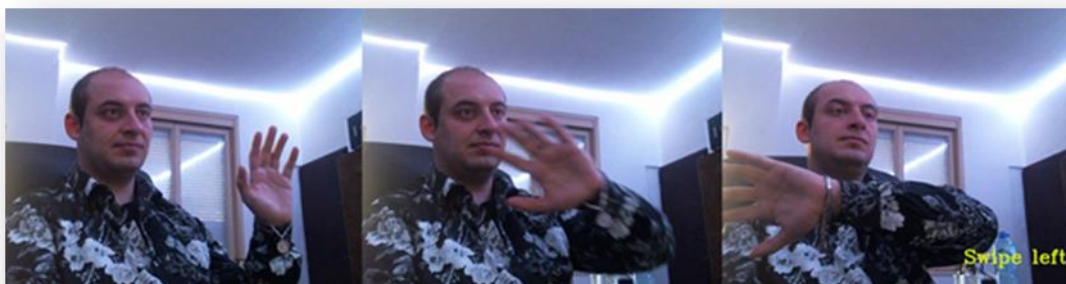
Основните характеристики на LSTM са показани в таблица 3.1

Таблица 3.1. Основни характеристики на LSTM

LSTM - основни характеристики	
Брой неврони	224 * 224 * 3 (150 528)
Pooling	Pool size=(7,7)
Flatten	
Dropout	0.5
Функция на активиране на изходния слой	Softmax

- разпознаване с определяне на начална и крайна позиция на жеста

Обучената ConvLSTM е базирана на MobileNet невронна мрежа и може да разпознава два типа жеста от различна дистанция – отместване на ръка в дясно и отместване на ръка в ляво



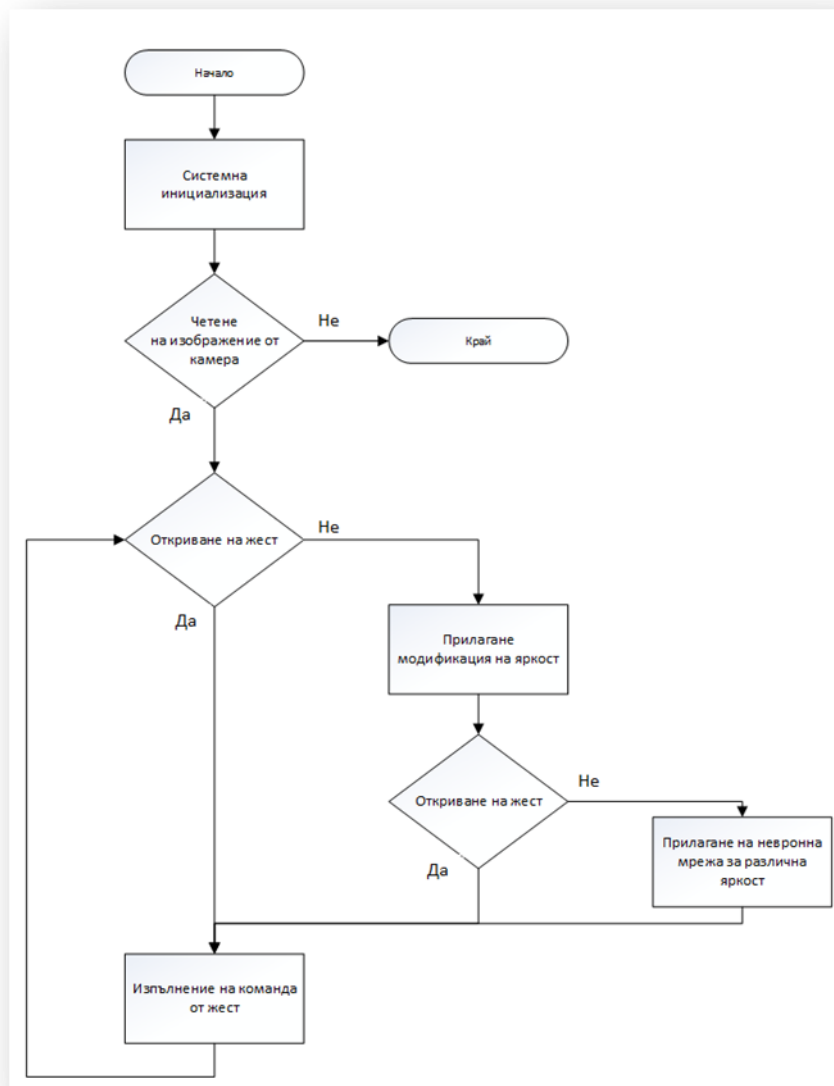
Фиг 3.42 Кадри от видео за откриване на ляв замах с ръка



Фиг 3.43 Кадри от видео за откриване на десен замах с ръка

3.4 Разпознаване на жестове при различна осветеност

В различни помещения има различна осветеност поради различни причини – брой осветителни тела, тяхната мощност и размер на помещението. Поради тази причина е необходимо софтуерно да се нормализират получените изображения използвайки различни подходи като динамичен избор на невронна мрежа или софтуерно променяне на яркостта.



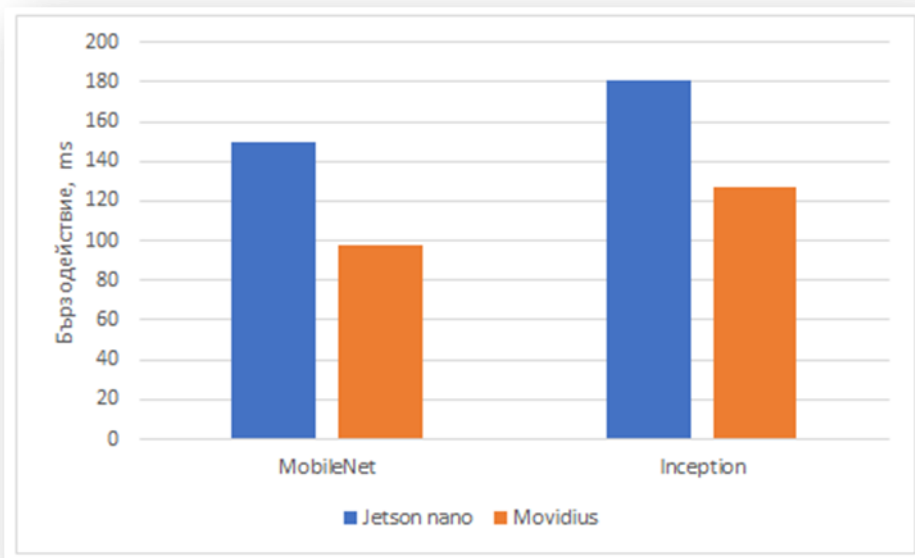
Фиг 3.47 Алгоритъм за откриване на жестове при различна осветеност

Програмата започва с инициализация и четене на изображения от камера. Ако изображението не е прочетено програмата спира своето действие. При успешно прочетена картина се прави опит за откриване на жест. При неуспешно открит жест се прави модификация на яркост и нов опит за неговото откриване. Ако след модификацията жест все още не е открит програмата сменя невронната мрежи с друга невронна мрежа с архитектура Inception.

Предложените алгоритми са тествани върху платформите Jetson Nano и Movidius с архитектурите R-CNN Inception и MobileNet тяхната точност е показана в таблица 3.2

Таблица 3.2. Бързодействие при избраните платформи

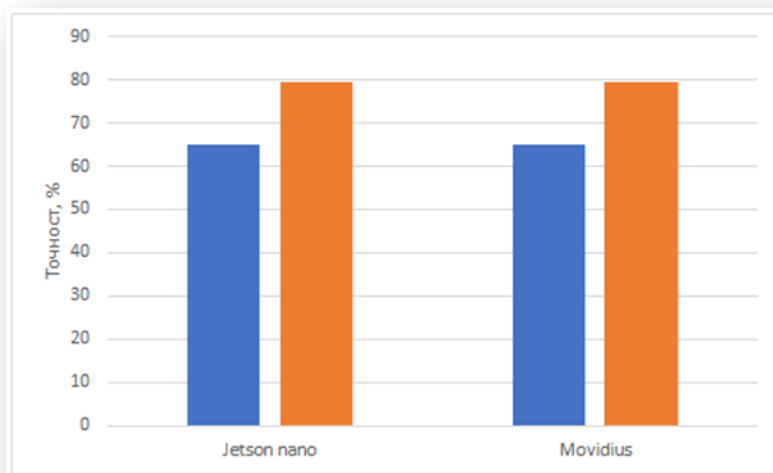
Хардуер	Бързодействие на обработката, msec	
	MobileNet	Inception
Jetson nano	150	181
Movidius	98	127



Фигура 3.48. Бързодействие при различни платформи
Точността при локализиране на жестове е показана на таблица 3.3

Таблица 3.3. Точност при избраните платформи

ОС	Точност при обработка, %	
	MobileNet	Inception
Jetson nano	79.3	65
Movidius	79.3	65



Фигура 3.49. Точност при избраните платформи

В таблица 3.4 са представени метрики за оценка на работата на MobileNet

Таблица 3.4. Резултати на архитектура MobileNet при мерки за надеждност

Мерка за надеждност	Резултат
Precision	0.642370215
Recall	0.769472362
F-measure	0.700200057

В таблица 3.5 са представени метрики за оценка на работата на Inception

Таблица 3.5. Резултати на архитектура Inception при мерки за надеждност

Мерка за надеждност	Резултат
Precision	0.759240759
Recall	0.805511394
F-measure	0.781691952

От направените изследвания се вижда, че за откриване на жестове Inception архитектурата дава по-добра точност, както и операционната система не оказва влияние върху нея.

От направените изследвания се вижда, че за откриване на жестове Inception архитектурата дава по-добра точност, както и операционната система не оказва влияние върху нея.

Изводи по глава III

- Разработен е алгоритъм за интерпретиране на жестове над човешката глава и изпълнение на различни видове команди.
- Разработен е алгоритъм за откриване на жестове и безопасна дистанция в индустриални системи.
- Предложен е адаптивен алгоритъм за откриване на жестове при различна осветеност.
- Използвана и тествана е LSTM невронна мрежа при различна осветеност за обработка на видео последователност.

ГЛАВА IV. Приложения на изкуствени невронни мрежи за откриване и локализиране на пожари

Използването на комбинацията машинно зрение и невронни мрежи за откриване на възникването на пожари е иновативен подход, посредством който се отстраняват някои от недостатъците на пожароизвестителните датчици и може да се увеличи акуратността при локализиране на пожари. Едно от приложенията на подобен подход е в роботизирани пожарогасителни системи и такова приложение е изследвано в дисертационния труд. За реализацията на алгоритмите на работа на пожарогасителната система, по време на изследванията е разработена роботизирана мобилна платформа.

4.1 Блокова схема на работата и използвани основни компоненти

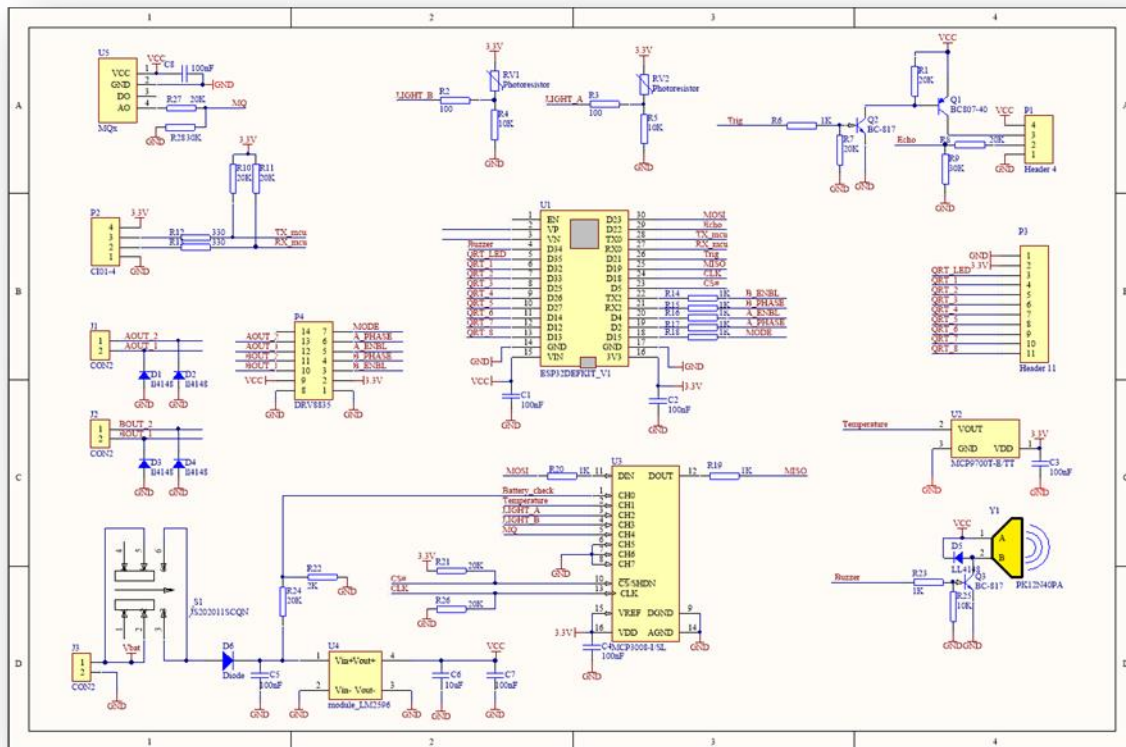
С оглед на характеристиките на съществуващи платформи на роботи, при дисертационните изследвания е реализиран мобилен робот на базата на JetsonNano със свързани към него HDMI дисплей, USB камера както и WiFi модул ESP32, който получава команди за движения при открит пожар от Jetson Nano освен това притежава и набор от периферни сензори свързани към него които могат да предават информация към главният хардуер.



Фиг. 4.13 Блокова схема на роботизираната платформа

Разработеният робот е базиран на платформата JetsonNano на nVidia. Системата се захранва от 12 V батерия. Тъй като с изключение на стъпковите мотори по-голямата част от хардуерното решение работи на 5 V се ползва импулсен преобразувател, който преобразува захранващото 12V напрежение.

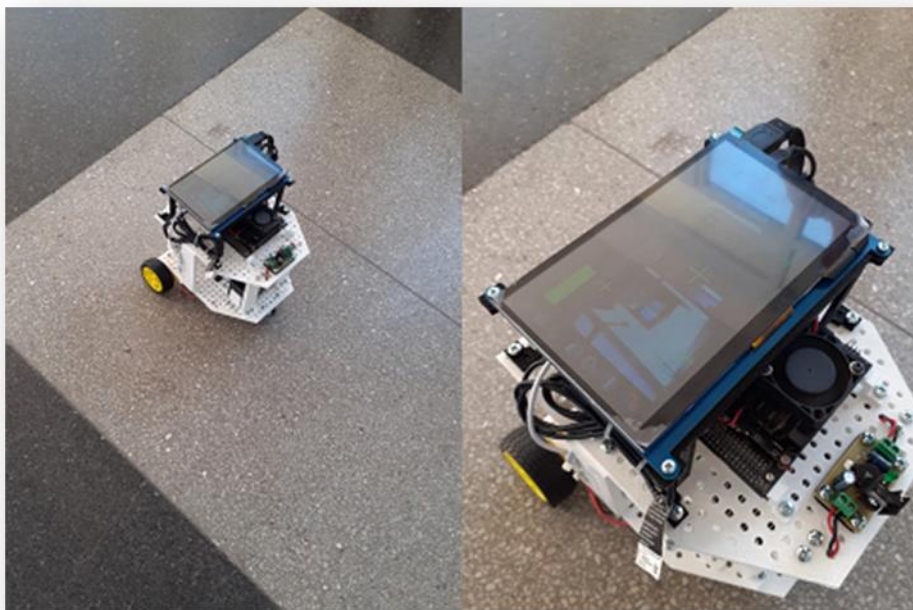
Платформата получава изображения от USB камера, които се подават към Jetson Nano модула и се визуализират на HDMI дисплей. При открит огън във входното изображение, Jetson Nano започва изпращането на команди към ESP32, съдържащи посока и обороти на двигателите за достигане до огъня, като се задвижват два постояннотокови мотора. Модулът LM2596 осигурява захранващо напрежение от 5V на по-голямата част от принципната схема с изключение на моторите които работят на 12V. Откритият огън от Jetson Nano се маркира и визуализира на HDMI дисплеят. Освен моторите, към периферията на ESP32 има свързан зумер за звукова индикация при открит пожар, температурен сензор който служи за спиране движението на робота ако около него е твърде горещо и ултразвуков сензор HC-SR04 за откриване на обекти които биха представлявали препятствие. Принципната схема на разработеният модул за управление е показана на фигура 4.14.



Фиг. 4.14 Принципна схема на Модул ESP32 – управление на сензори и двигатели

4.2 Външен вид на роботизираната платформа

В предложената конструкция на робота е включен HDMI дисплей, на който може да се наблюдава обработената информация от камерата, като под дисплея е разположена платката Jetson Nano както и стабилизатора за захранващото напрежение. Под тях са позиционирани ESP микроконтролерът със свързаните към него сензори, мотори, както и 12V батерия за захранване на робота. При извършените тестове системата притежава добри динамични характеристики върху равни повърхности.



Фиг. 4.15 Роботизирана платформа с Jetson Nano модул

4.3 Комуникация между Jetson Nano и ESP32 - Сериен порт RS-232

- Управление на задвижването

Разработеният формат на командите за управлението на стъпковите мотори се подават в следният вид:

`?AF:2000\r\n`

Където :

? – тип на команда (? – мотор, ! – сензор)

A – кой мотор да се задвижва (L – ляв, R – десен, A – двата)

F – посока на въртене (F – напред, B – назад)

: - параметър за продължителност. Числото 2000 означава, че текущата команда ще се изпълнява в продължение на 2000 ms.

\r\n – край на команда

Команда	Описание
<code>?RF:1000\r\n</code>	Десен мотор напред в продължение от 1000 ms
<code>?LF:1000\r\n</code>	Ляв мотор напред в продължение от 1000 ms
<code>?RB:1000\r\n</code>	Десен мотор назад в продължение от 1000 ms
<code>?LB:1000\r\n</code>	Ляв мотор назад в продължение от 1000 ms
<code>?AF:1000\r\n</code>	Двата мотор напред в продължение от 1000 ms
<code>?AB:1000\r\n</code>	Двата мотор назад в продължение от 1000 ms

Таб 4.2 с възможни команди за управление на мотор

В зависимост от командата управляващи импулси се изпращат към драйвера DRV8835 който включва / изключва DC мотора и определя неговата посока.

- Прочитане на сензорите

Освен за управление на постояннотоков мотор, съществуват команди и за четене на сензорите. Проектираният хардуер съдържа два сензора, ултразвуков сензор и сензор за температура, примерна команда за четене е следната:

`!T\r\n`

! – команда за сензор

T – четене на температурен сензор

\r\n – край на команда

Команда	Описание
<code>!S\r\n</code>	Четене на ултразвуков сензор
<code>!T\r\n</code>	Четене на сензор за температура

Таб 4.2 с възможни команди за четене на сензор

4.4 Тестване на различни DNN за откриване на пожара посредством роботизираната платформа

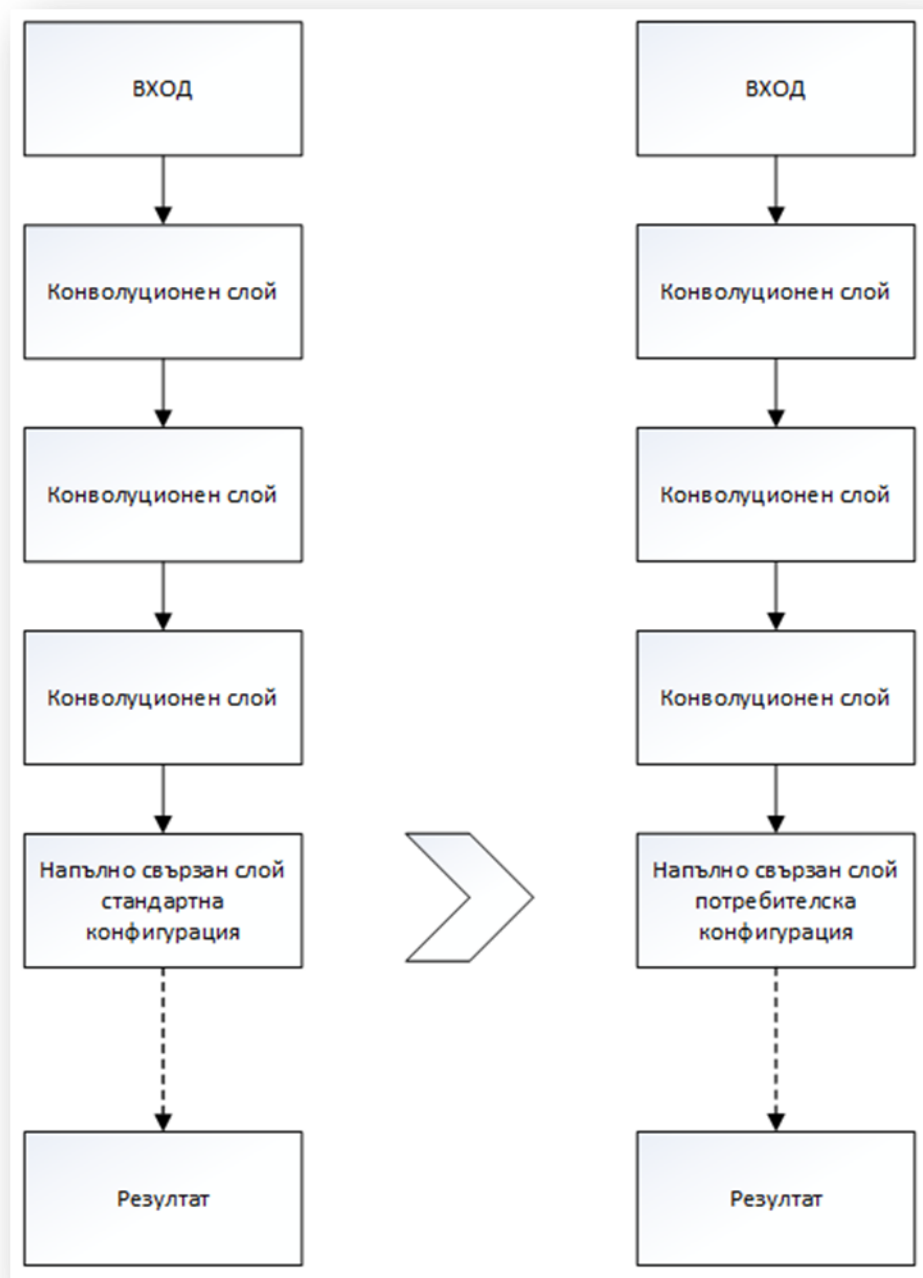
В тестовите са използвани невронни мрежи, които имат следните предимства и недостатъци на архитектурите;

- SSD MobileNet - Има високо бързодействие, но ниска точност при търсене на малки обекти сравнение с други архитектури. При търсене на големи обекти SSD може да притежава по-висока точност от R-CNN;
- Mask R-CNN – Базирана на R-CNN тя притежава възможността освен да връща локацията на обект и да прилага маска върху неговите пиксели. За разлика от SSD MobileNet и Yolo тази невронна мрежа притежава сравнително по-високо време за реакция;
- Yolo – Предимството е, че при нея липсва плъзгащ се прозорец и цялата се разглежда като един цял регион. Притежавайки високо бързодействие и точност, тя не е толкова ефективна при множество открити обекти в изображението, които са много близко един до друг, в резултат на което не всички биват локализирани в крайният резултат.

4.4.5.4 Предварителна обработка на данните от камерата чрез филтрация и модификации на мрежите за локализиране на пожари

Преди да се приложат данните за обучение към невронната мрежа те биват събрани от различни източници (например от камера) където е необходима съответната филтрация [139]. Тя се състои в преоразмеряването на снимките в необходимия формат за входа на невронната мрежа. Колкото по-голяма е базата данни толкова по-висока точност ще има невронната мрежа. При събирането на данни трябва да се вземе под внимание разстояние, осветеност, размер на обекта, фон и други. Желателно е всички изображения да бъдат съхранявани в еднакъв формат който да съответства на входния слой на проектираната невронна мрежа.

За обучението на невронните мрежи е използвано метода Transfer Learning [140]. Вместо да се използва изолация се използва информацията придобита от една задача за решаване на сходни задачи. При обучението се взема модел на невронна мрежа, която притежава множество слоеве за разпознаване на голям брой класове. Последните слоеве, които са напълно свързани слоеве в зависимост от броя на класовете, се изрязват и се заменят с нов напълно свързан слой, който включва характеристиките за откриване на огън. Фиг 4.18 обяснява визуално посоченият процес



Фиг 4.18 Премахване на ненужни слоеве при трансферно обучение

Използва се ReLu функция за активиране, която на изхода си изкарва входните данни ако те са положителни, в противен случай изхода е нулев. Обичайна практика е да се извлече активирането на напълно свързаните слоеве преди ReLu, т.е. с отрицателни стойности, в случаите, когато липсват априорни знания за това кои визуални класове може да има в изображението.

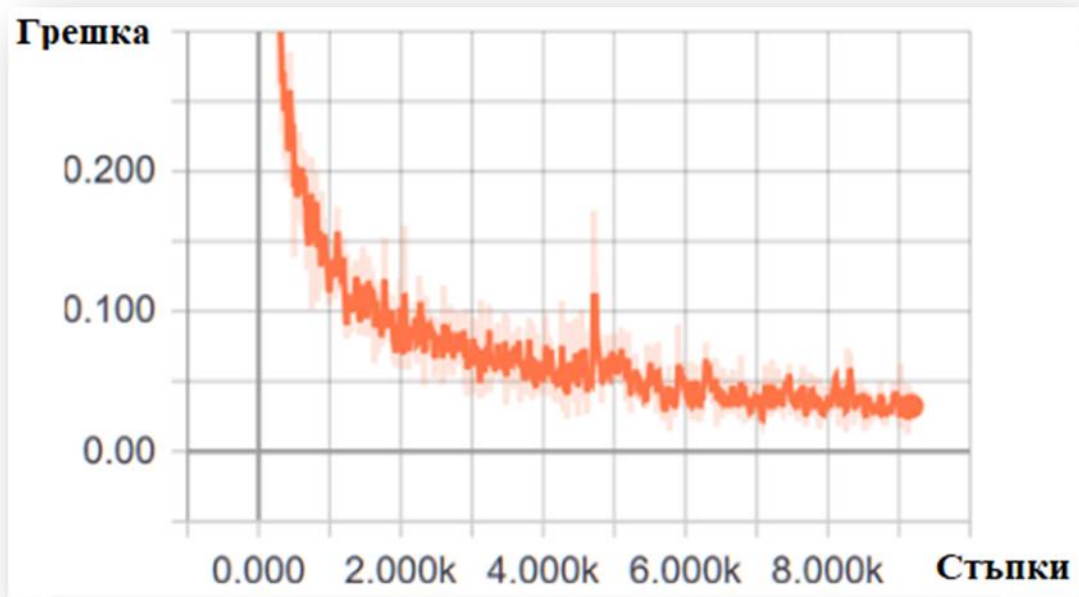
Посочените действия намаляват размера на невронна мрежа (от около 20 Mb на ~ 5 Mb) оптимизирайки я да разпознава желаните класове с по-високо бързодействие, освен това намаляването на размера позволява нейното интегриране в устройства с ограничен обем памет.

4.5 Обучение на невронните мрежи

- SSD MobileNet и Mask-RCNN

Невронна мрежа е обучена, използвайки споменатите архитектури SSD MobileNet, Mask-RCNN. Обучението изисква предварителна база данни от изображения, от които е

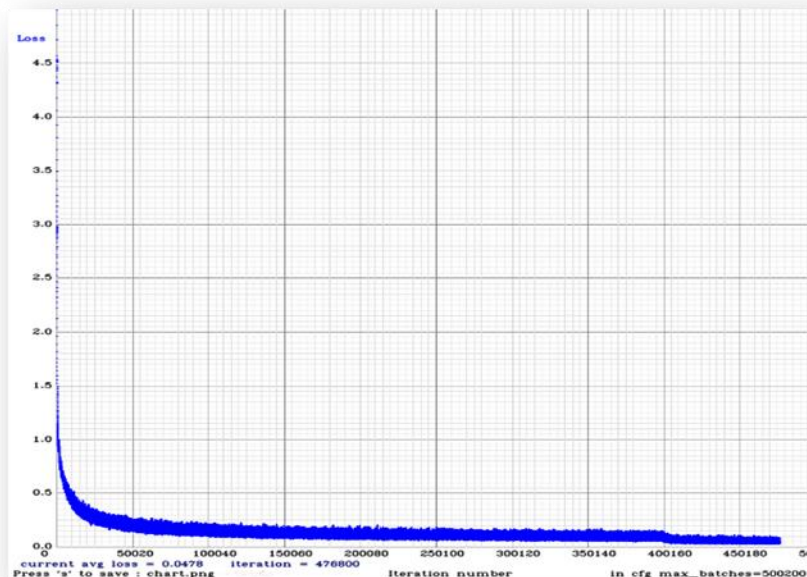
необходимо да се извлекат характеристиките на търсените обектите. При началната обработка е необходимо всички изображения, предвидени за нуждите на обучението да бъдат с размер, съответстващ на входния слой на невронната мрежа. При разработената система е избран стандартен размер 300x300 за MobileNet и 800x600 за Mask-RCNN. С използването на инструмент за маркировка в процеса на предварителната обработка на обучаващите данни, се извършва локализация на координатите на търсения обекта в изображението и му се присвоява етикет(т.нар. labeling). За Mask-RCNN има една особеност, че е необходимо да се създадат т.нар. анотации. Анотациите представляват пикселно означаване със специфичен цвят, указващи че там е желаният обект, който може да бъде във всякаква форма. От етикетите и анотациите се създава “tfrecord” формат на базите данни, който е необходим за тренирането на невронната мрежа. За нейното обучение е необходимо да се конфигурират броят класове за разпознаване на невронната мрежа, размера (batch size) на броя изображения, които ще се подават на невронната мрежа при всяка итерация, както и база данни за трениране на невронната мрежа и тестване на грешката. Използвайки Tensorflow при обучението е възможно следене на текущото състояние и визуализацията му с помощта на инструмента Tensorboard. По този начин се предоставя възможност в реално време да се визуализират текущите грешки в процеса на обучението (фиг.4.19).



Фиг 4.19 Обучение на MobileNet и Mask R-CNN в Tensorboard

- Yolo

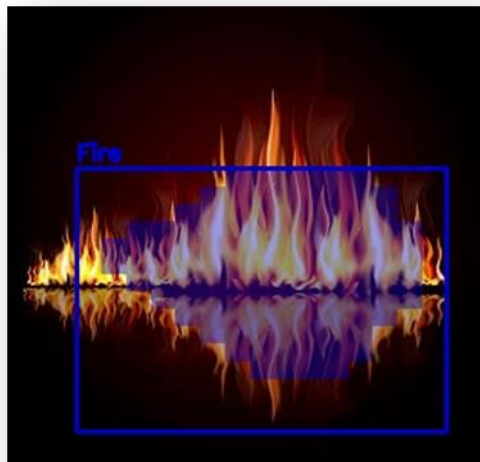
Тук има няколко основни различия при обучението спрямо това при MobileNet и Mask R-CNN. Първо входния слой приема изображения с различна резолюция, причината за това е, че тук липсва така нареченият „плъзгащ прозорец“ и е избрана резолюция 416x416. При локализацията на изображенията в базата данни те се запазват в различен формат. До като предходните архитектури запазваха формата в стандартен вид (x,y, ширина, височина) тук се влиза под внимание центъра на обекта и неговите крайни точки (център x, център y, ширина и височина). Останалата методика за тренирането на невронната мрежа е същата като предходните.



Фиг 4.20 Обучение на невронна мрежа Yolo

4.6 Тестване на работата на обучените невронни мрежи

Обучените невронни мрежи могат да открият обекти в различни ситуации. Тествани са със следните изображения, които не са включени по време на обучението. На фигура 4.21 е показано откриване на пожар в обикновено изображение за основно тестване, без други обекти в него, използвайки Mask R-CNN.



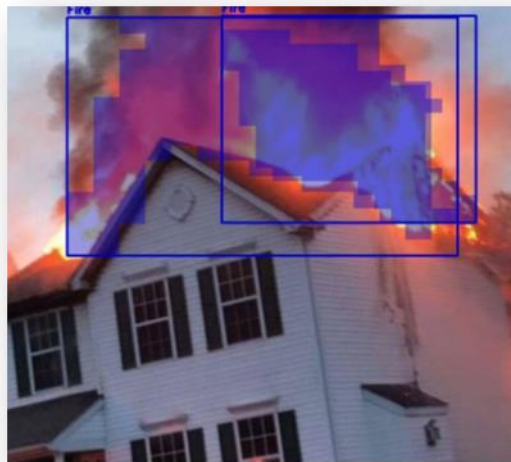
Фиг. 4.21 Откриване на огън с MASK R-CNN

За втория тест на Mask R-CNN мрежа е избрано изображение с пожар в помещение и наличие на други предмети с подобна на огъня сигнатура. Резултатът може да се види на фиг 4.22.



Фиг. 4.22 Откриване на огън в стая

За третия тест е използвана снимка на горящ покрив при дневна светлина и резултатите могат да се видят по-долу.



Фиг 4.23 Открит огън върху горяща сграда

4.6 Алгоритъм за откриване на действителен пожар и на статична картина с огън

Първоначално за тестване на точността на невронната мрежа за откриване на огън е реализиран алгоритъм, при който се четат кадри от камера, които се подават на невронната мрежа, която от своя страна връща местоположението на огъня, ако е открито такова, както и процента за сигурност дали това наистина е търсеният обект. Разработен е алгоритъм, който проверява дали огънят е статичен или истински. Той работи като в последователност от прочетени кадри координатите на огъня се следят дали се променят. Тестовият алгоритъм е показан на фиг. 4.24.



Фиг. 4.24 Алгоритъм за откриване и различаване на статичен и динамичен огън

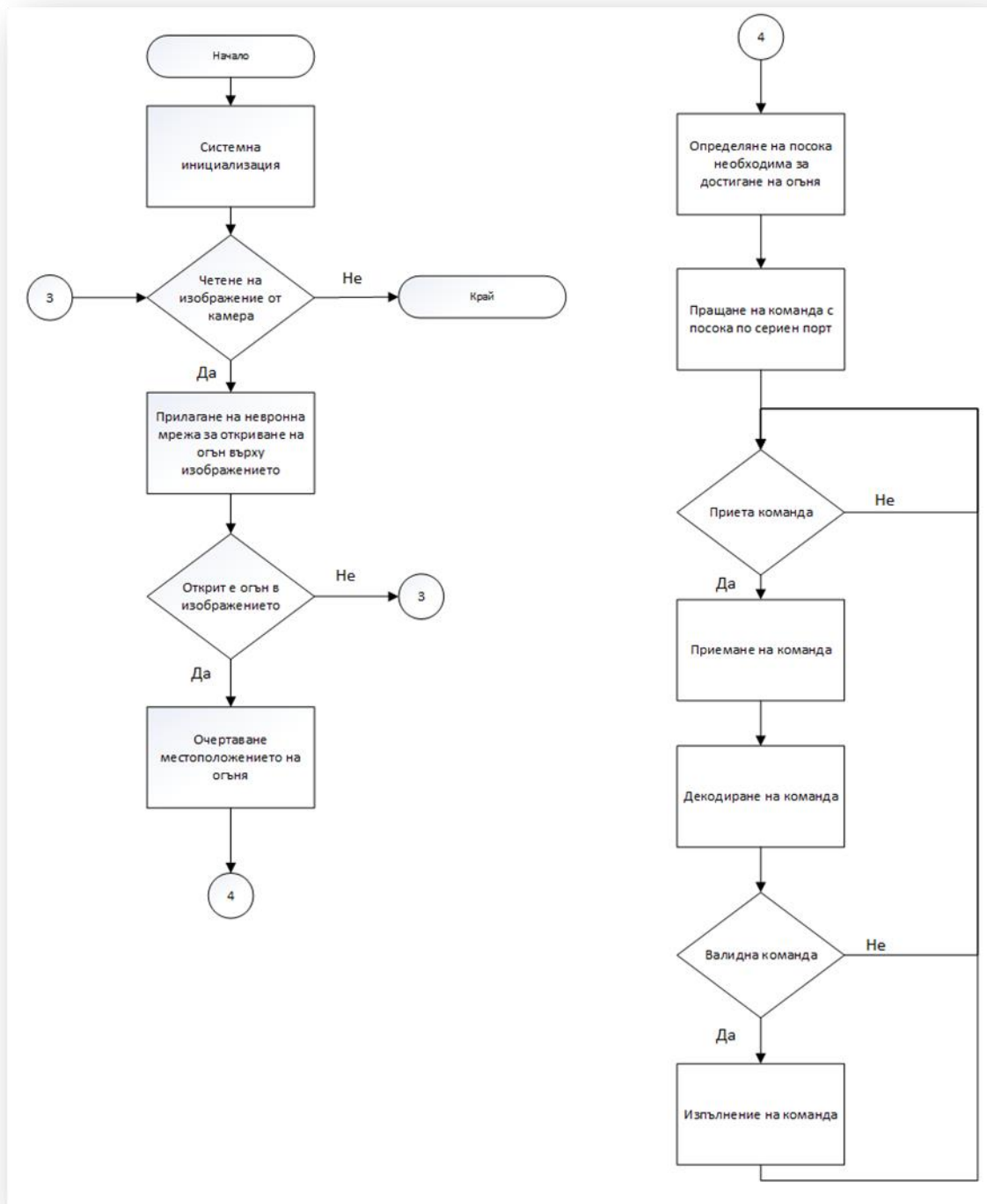
След тестването и установяването на висока надеждност с използваната невронна мрежа се интегрира в алгоритъм за динамични изображения. Текущите резултати са показани на фиг. 4.25.



Фиг. 4.25 Резултати при откриване на огън

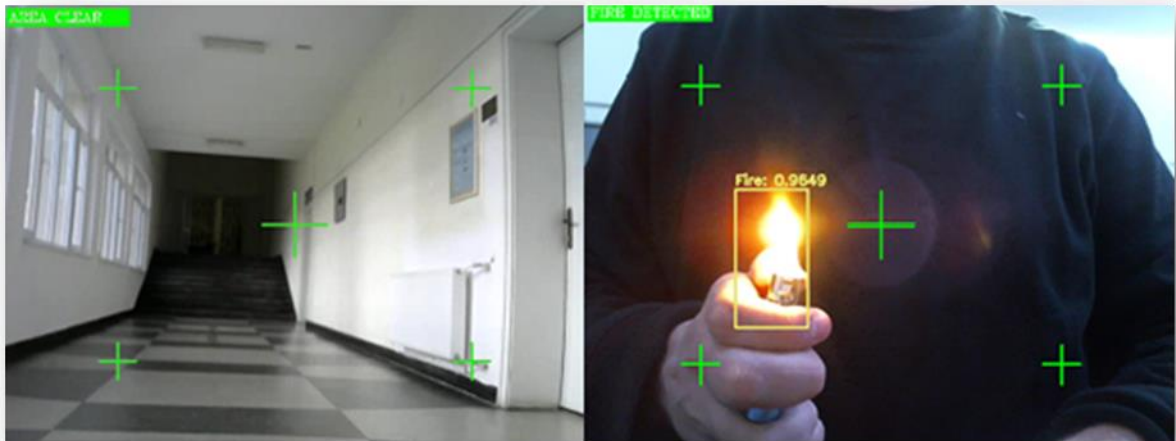
4.7 Интегриране на обучените невронни мрежи и тестване на работата им в роботизираната платформа

Реализиран е тестови робот, който симулира автономен пожарогасител за акустично гасене на пожар. В мобилния робот е имплементирана тестваната Yolo невронна мрежа. При стартиране на захранването на робота се инициализира операционната система на Jetson Nano, стартира се програмата за управление, както се стартира и програмата на WiFi Модулът ESP32. Робота постоянно следи за пожар сканирайки входните изображения от камерата прилагайки невронната мрежа. Веднъж открит огън в изображението, неговото местоположение се показва върху HDMI дисплей с очертание на неговите координати, име на клас и процент на сигурност на разпознаване. Спрямо позиционирането на огъня се изчислява какви движения са необходими за неговото достигане. Имайки необходимите данни за достигане до огъня те се пращат по сериен порт към WiFi модулта ESP32, който от своя страна ги декодира и изпълнява връщайки резултата от командата. WiFi модулта също така активира звукова индикация при открит пожар и следи сензор за температура. При отчитане на температура над поставеният праг робота спира да се движи. Като робота е на безопасно разстояние в позиция на гасене, той може да активира акустична система за гасене на пожар, също така ако WiFi модулта е свързан към мрежа да изпраща данни за текущата ситуация. Алгоритъмът за управление е показан на фиг 4.29.



Фиг. 4.29 Алгоритъм за откриване на опасност интегриран в робот база на Jetson Nano

При тестването на алгоритъма се установява, че предложената система е работоспособна и може да бъде използвана също и за обучителни цели. Снимки, направени от камерата на робота са показани на фиг 4.30.



Фиг. 4.30 Изглед от работа

В таблица 4.4 са представени метрики за оценка на работата на MobileNet

Таблица 4.4. Резултати на архитектура MobileNet при мерки за надеждност

Мерка за надеждност	Резултат
Precision	0.863636364
Recall	0.863636364
F-measure	0.863636364

В таблица 4.4 са представени метрики за оценка на работата на Mask R-CNN

Таблица 4.5. Резултати на архитектура Mask R-CNN при мерки за надеждност

Мерка за надеждност	Резултат
Precision	0.931372549
Recall	0.931372549
F-measure	0.931372549

От направените изследвания се вижда, че Mask R-CNN дава по-добри резултати при откриването на пожар. Разработената невронна мрежа може да бъде използвана в различни приложения и вградени системи където е необходимо да бъде открит пожар и може да бъде доразвита разпознавайки и други видове бедствия като течове и дим.

Изводи по глава IV

- Разгледани са различни видео противопожарни системи и е разработен интелигентен безжичен сензор с интегрирана невронна мрежа за откриване на пожар. Сензора е предназначен за интегриране в различни пожароизвестителни и пожарогасителни системи за обществени, промишлени и фамилни сгради и помещения.
- Проектирани са невронни мрежи с различни архитектури за откриване на пожар на базата на които са разработени и тествани приложения в различни VPU и GPU модули.
- Модифицирани са DNN невронни мрежи променяйки техните слоеве с цел оптимизиране на работата им при откриването на пожар.
- Изследвана е точността и бързодействието при откриване на пожар от сензори с различни архитектури на невронната мрежа;
- Разработени и тествани са различни алгоритми за откриване на пожар и е предложен алгоритъм за гасене на пожар интегриран в автономен робот.
- Предложена е архитектура на съвременна домашна система за откриване на пожар с висока надеждност и бързодействие.

НАУЧНО-ПРИЛОЖНИ ПРИНОСИ НА ДИСЕРТАЦИОННИЯ ТРУД

1. Предложен е адаптивен алгоритъм за откриване на агресия на базата на конволюционни невронни мрежи, който е реализиран и тестван в реална среда;
2. Предложен е адаптивен алгоритъм за откриване и разпознаване на жестове при различна осветеност;
3. Предложен е адаптивен алгоритъм за разпознаване на динамични жестове;
4. Предложен е алгоритъм за разпознаване на огън и са направени изследвания при използване на алгоритъма с различни модели изкуствени невронни мрежи.

ПРИЛОЖНИ ПРИНОСИ НА ДИСЕРТАЦИОННИЯ ТРУД

1. Реализирано е мобилно приложение за откриване на опасни предмети в изображения от IP камера;
2. Реализиран е интелигентен сензор за откриване на пожар, който може да бъде интегриран в различни помещения;
3. Реализиран е интелигентен робот за откриване на пожари с помощта на изкуствени невронни мрежи.

СПИСЪК НА ПУБЛИКАЦИИ ПО ДИСЕРТАЦИОННИЯ ТРУД

1. Ivanov, S., S. Stankov, T. Nenov. FPGA Based Neural Networks for Characters Recognition. 20th International Symposium on Electrical Apparatus and Technologies (SIELA), June 3-6, 2018, Bourgas, 2018, pp. 1-4, IEEE Xplore (SCOPUS), Electronic ISBN: 978-1-5386-3419-6, doi: 10.1109/SIELA.2018.8446658
2. Stankov S., S. Ivanov, Gesture command in home environment, Unitech'2019, 15-16 Nov 2019, Gabrovo, pp.I-248 – I-251
3. Stankov S., S. Ivanov, Finding fire with the help of artificial neural networks, Международна научна конференция „Изкуствен интелект и Е-лидерство” 10-11.10.2019 г.
4. Stankov S., S. Ivanov, T.Todorov, An application of deep neural networks in industrial robotics for detection of humans - Proc. XXVIII International Scientific Conference Electronics - ET2019, September 12 - 14, 2019, Sozopol, Bulgaria, (Scopus)
5. Stefan Ivanov, Stanko Stankov, Jacek Wilk-Jakubowski, Pawel Stawczyk, The Using of Deep Neural Networks and Acoustic Waves Modulated by Triangular Waveform for Extinguishing Fires, International Workshop on New Approaches for Multidimensional Signal Processing, NAMSP 2020, Sofia, Bulgaria July 09-11, 2020
6. Stankov S., S. Ivanov, INTELLIGENT SENSOR FOR FIRE DETECTION WITH DEEP NEURAL NETWORKS, JOURNAL OF INFORMATICS AND INNOVATIVE TECHNOLOGIES (JIIT), брой 1, 2020, ISSN: 2682-9517, pp.25-28
7. Stankov, Stanko, and Stefan Ivanov. "Finding anomalies with artificial neural network." Industry 4.0 4.3 (2019): 128-129.
8. Stankov S., S. Ivanov, SYSTEM FOR IMAGE RECOGNITION BASED ON RASPBERRY PI, International Scientific Conference Unitech, 2018, 16-17 November 2018, GABROVO
9. S. Stankov, FINDING DANGEROUS OBJECTS WITH THE HELP OF ARTIFICIAL NEURAL NETWORKS, International Conference "Automatics and Informatics 2019", 03-04.10.2019

10. Stankov S., S. Ivanov, APPROXIMATION OF FUNCTION THROUGH A HARDWARE REALIZED NEURAL NETWORK, INTERNATIONAL SCIENTIFIC CONFERENCE 17-18 November 2017, GABROVO
11. Stankov S., S. Ivanov, MOBILE ROBOT FOR FIRE DETECTION, INTERNATIONAL SCIENTIFIC CONFERENCE 20-21 November 2020, GABROVO
12. Stefan Ivanov, Stanko Stankov, ACOUSTIC EXTINGUISHING OF FLAMES DETECTED BY DEEP NEURAL NETWORKS IN EMBEDDED SYSTEMS, 6th International Conference on Smart City Applications, 27-29 October 2021, Turkey

ЦИТИРАНИЯ НА ПУБЛИЦИИ ПО ДИСЕРТАЦИОННИЯ ТРУД

Публикация:	Цитирана в:
<p>1. Ivanov S, Stankov S, Nenov T. FPGA Based Neural Networks for Characters Recognition. In 2018 20th International Symposium on Electrical Apparatus and Technologies (SIELA) 2018 Jun 3 (pp. 1-3). IEEE.</p>	<p>1. Weifei F. Intelligent recognition of motion posture based on FPGA and neural network. <i>Microprocessors and Microsystems</i>. 2020 Nov 4:103374.</p> <p>2. Aydin SG, Bilge HŞ. FPGA based implementation of sigmoid function using different approaches. In 2020 28th Signal Processing and Communications Applications Conference (SIU) 2020 Oct 5 (pp. 1-4). IEEE.</p> <p>3. Du, Nan, Heidemarie Schmidt, and Iliia Polian. "Low-power emerging memristive designs towards secure hardware systems for applications in internet of things." <i>Nano Materials Science</i> (2021).</p>
<p>2. Stankov S, Ivanov S, Todorov T. An Application Of Deep Neural Networks In Industrial Robotics For Detection Of Humans. In 2019 IEEE XXVIII International Scientific Conference Electronics (ET) 2019 Sep 12 (pp. 1-3). IEEE.</p>	<p>4. Афанасьев МЯ, Федосов ЮВ, Крылова АА, Шорохов СА. Применение машинного зрения в задачах автоматического позиционирования инструмента модульного оборудования. <i>Известия высших учебных заведений. Приборостроение</i>. 2020;63(9). (2020).</p> <p>5. Afanasev MY, Fedosov YV, Andreev YS, Krylova AA, Shorokhov SA, Zimenko KV, Kolesnikov MV. Machine Vision for Auto Positioning in the Modular Industrial Equipment: A Case Study. In 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS) 2020 Jun 10 (Vol. 1, pp. 223-229). IEEE.</p>
<p>3. Stefan Ivanov, Stanko Stankov, Jacek Wilk-Jakubowski, Paweł Stawczyk, The using of Deep Neural Networks and acoustic waves modulated by triangular waveform for extinguishing fires, <i>New Approaches for Multidimensional Signal Processing</i>, Springer, Singapore, 2021, pp. 207-218</p>	<p>6. Shevchenko, R. I., Strelets, V. M., Loboichenko, V. M., Pruskyi, A. V., Myroshnyk, O. N., & Kamyshentsev, G. V. (2021). Review of up-to-date approaches for extinguishing oil and petroleum products.</p> <p>7. Taspinar, Y. S., Koklu, M., & Altin, M. (2021). Classification of flame extinction based on acoustic oscillations using artificial intelligence methods. <i>Case Studies in Thermal Engineering</i>, 28, 101561.</p>